

Ingeniería en Tecnologías Industriales 2018-2019

Trabajo Fin de Grado

DESARROLLO DE UN SIMULADOR DE INCENDIOS FORESTALES. INFLUENCIA DEL VIENTO CON CFD

Ignacio Zamanillo Herreros

Tutor:

Dr. Santiago Garrido Bullón

10 junio, 2019



[Incluir en el caso del interés de su publicación en el archivo abierto]

Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**

Resumen

Los incendios forestales son un problema que debe afrontar la sociedad actual pues producen daños humanos y materiales irreparables.

Una de las mayores dificultades en la extinción de incendios es prever la dirección del viento, ya que este es uno de los principales factores para la extensión del incendio. El objetivo de este Trabajo de Fin de Grado es desarrollar una herramienta capaz de calcular la dirección y velocidad del viento con el objetivo de evitar la propagación del fuego.

Para ello, se hará uso del programa **Matlab** y su app **CFD Toolbox for MATLAB** que nos permitirán realizar simulaciones para:

- Utilizar mapas reales.
- Elegir la dirección del incendio y la velocidad de este.
- Visualizar la trayectoria del viento en mapas 3D.

Hay que señalar que este TFG forma parte de la línea de investigación abierta en esta universidad, cuyo objetivo es determinar qué técnica realiza una mejor predicción.

Palabras clave: viento, incendios, CFD (Dinámica de fluidos computacional) y Matlab.

Agradecimientos

A mi familia, por todo el apoyo que me han dado durante el grado.

A mi amigo Pablo, por su ayuda y paciencia y por hacerme reír en todo momento.

A Cristina, por su cariño en estos largos meses.

A Santiago por ayudarme con sus acertadas aportaciones a realizar este TFG, por su paciencia y por mostrarme infinidad de áreas de conocimiento.

Índice

1. Introducción	1
1.1 Motivación	1
1.2 Datos sobre incendios en España.....	1
1.2.1 Datos generales.....	1
1.2.2 Causas de los incendios en España	3
1.2.3 Consecuencias de los incendios forestales	4
1.2.4 Medios para evitar los incendios y sus daños.....	4
1.3 Objetivo.....	5
1.4 Estructura	6
2. Estado del arte.....	7
2.1 Modelos físicos.....	7
2.2 Modelos empíricos.....	8
2.3 Modelos de Simulación Computacional de Incendios (MSCI).....	9
2.4 Modelos basados en el algoritmo de Fast Marching	11
2.5 Principales programas de predicción	12
2.5.1 FireRS.....	12
2.5.2 Prometheus	13
2.5.3 FARSITE.....	15
2.5.4 Pyrowarn	17
3. Desarrollo del trabajo	18
3.1 Conceptos teóricos	18
3.2 Mapas de elevación del terreno.....	20
3.2.1 Prueba con un mapa de elevación ficticio	20
3.2.2 Prueba con un mapa de elevación real	20
3.2.3 Muestra de mapas reales	23
4. Muestra de resultados	27
4.1 Muestra de trayectorias en la superficie generada por Peaks.....	27

4.2 Muestra de trayectorias en terrenos reales.....	28
5. Validación del modelo.....	32
5.1 Tesis de Jason M. Forthofer	32
5.1.1 Condiciones de contorno	32
5.1.2 Mallado.....	32
5.1.3 Askervein hill	32
5.2 Comparación de los datos obtenidos por Forthofer con los del TFG.....	35
5.2.1 Generar un polinomio aproximado.....	37
5.2.2 Generar el mallado y definir las superficies	37
5.2.3 Resolución y comparativa de los resultados	39
5.3 Comparativas:	39
5.3.1 Línea A	39
5.3.2 Línea AA.....	44
5.3.3 Línea B	48
6. Conclusiones: análisis y mejoras	52
6.1 Conclusión	52
6.2 Análisis crítico de los resultados	52
6.3 Entorno socioeconómico.....	53
6.4 Marco regulador.....	54
6.5 Mejoras y trabajo futuro	54
Bibliografía y Webgrafía	56
Anexos: Código.....	59
A.1 Código de Peaks	59
A.2 Código para realizar pruebas en una montaña real.....	62
A.3 Código usado en las pruebas con Forthofer	65
A.4 Código usado para comprobar las velocidades	69
A.5 Resto de funciones utilizadas.....	69

Índice de figuras

Fig 1.1- Superficie quemada en España	1
Fig 1.2- Incendios por término municipal en la Península Ibérica (2001-2010).....	2
Fig 1.3- Distribución mensual de los incendios en 2017	2
Fig 1.4- Principales causas de los incendios en España.....	3
Fig 2.1- Ejemplo de modelo de zona.....	9
Fig 2.2- Simulación de un incendio mediante el uso de un programa CFD.....	10
Fig 2.3- Propagación de una onda en diferentes campos de velocidades.....	11
Fig 2.4- Esquema del FIRE RS	12
Fig 2.5- Frente de onda mediante el principio de Huygens.....	13
Fig 2.6- Esquema con los inputs introducidos	14
Fig 2.7- FARSITE.....	15
Fig 2.8- Esquema de funcionamiento de FARSITE.....	16
Fig 2.9- Ejemplo de simulación de Farsite.....	17
Fig 3.1- Ejemplo de mapa de elevación ficticio, Peaks	20
Fig 3.2- Selección del Estado	21
Fig 3.3- Selección de los datos topográficos	22
Fig 3.4- Modificación del formato	22
Fig 3. 5- Selección del formato	22
Fig 3. 7- Código de importación de un mapa.....	23
Fig 3.8- Coordenadas: N 30.59073 y W -99.37331	23
Fig 3.9- Coordenadas: N 44.43274 y W -102.6587	24
Fig 3.10- Coordenadas: N 47.28841 y W -111.51761	24
Fig 3.11- Coordenadas: S 34.24644 y E -97.25181	25
Fig 3.12- Diagrama de flujo del proceso.....	26
Fig 4.1- Visualización de la trayectoria del viento utilizando Peaks.....	27
Fig 4.2- Visualización de la trayectoria del viento utilizando Peaks (2).....	27

Fig 4.3- Visualización de la trayectoria del viento utilizando Peaks (3).....	28
Fig 4.4- Coordenadas: N 30.59073 y W -99.37331	28
Fig 4.5- Coordenadas: N 30.59073 y W -99.37331	29
Fig 4.6- Coordenadas: N 44.43274 y W -102.6587	29
Fig 4.7- Coordenadas: N 47.28841 y W -111.51761	30
Fig 4.8- Coordenadas S 34.24644 y E -97.25181	30
Fig 4.9- Coordenadas: S 34.24644 y E -97.25181	31
Fig 5.1- Mapa en el que se enseña la posición de las líneas.....	33
Fig 5.2- Velocidad a lo largo de la línea A	34
Fig 5.3- Velocidad a lo largo de la línea AA.....	34
Fig 5.4- Velocidad a lo largo de la línea B.....	35
Fig 5.5- Diagrama de flujo de la comprobación de resultados.....	36
Fig 5.6- Ejemplo de superficie definida	37
Fig 5.7- Ejemplo del mallado	38
Fig 5.8- Mallado con superficies ya definidas	38
Fig 5.9- Superficie de la función de puntos A-A	40
Fig 5.10- Mallado de A	40
Fig 5.11- Mallado con superficies de A	41
Fig 5.12- Perfil de velocidades de A	41
Fig 5.13- Velocidad a lo largo de la línea A	42
Fig 5.14- Función de puntos AA.....	44
Fig 5.15- Mallado de AA	45
Fig 5.16- Mallado con superficies de AA	45
Fig 5.17- Perfil de velocidades de AA	46
Fig 5.18- Velocidad a lo largo de la línea AA.....	46
Fig 5.19- Función de B.....	48
Fig 5.20- Mallado de B	49

Fig 5.21- Mallado con superficies de B	49
Fig 5.22- Perfil de velocidades de C	50
Fig 5. 23- Velocidad a lo largo de la línea B.....	50

Índice de tablas

Tabla 5.1- Puntos utilizados para sacar la función.....	39
Tabla 5.2-Resumen de los resultados y errores.....	43
Tabla 5.3- Puntos utilizados para sacar la función.....	44
Tabla 5.4- Resumen de los resultados y errores.....	47
Tabla 5.5- Puntos utilizados para sacar la función.....	48
Tabla 5.6- Resumen de los resultados y errores.....	51

1. Introducción

1.1 Motivación

Periódicamente se publican noticias sobre incendios fuera de control, miles de hectáreas arrasadas e innumerables pérdidas, que afectan especialmente al medio ambiente. España es el país europeo al que más han afectado los incendios y el cuarto mundial que más superficie forestal ha perdido en lo que llevamos de siglo. Por ello es vital tener la capacidad de entender su propagación, para poder limitar los daños una vez se ha producido el incendio.

1.2 Datos sobre incendios en España

1.2.1 Datos generales (ver [1], [22])

En la Fig 1.1 se puede observar el número de hectáreas quemadas en España desde 1961 hasta 2014. Se puede observar la gran cantidad de superficie quemada entre los años 1980 y 1996, siendo 1985 el año con mayor superficie quemada desde que se poseen datos. El descenso de hectáreas afectadas se debe a la mejora de los servicios de extinción, ya que el número de incendios no ha disminuido significativamente. El peor dato de incendios de la última década tuvo lugar en 2012.

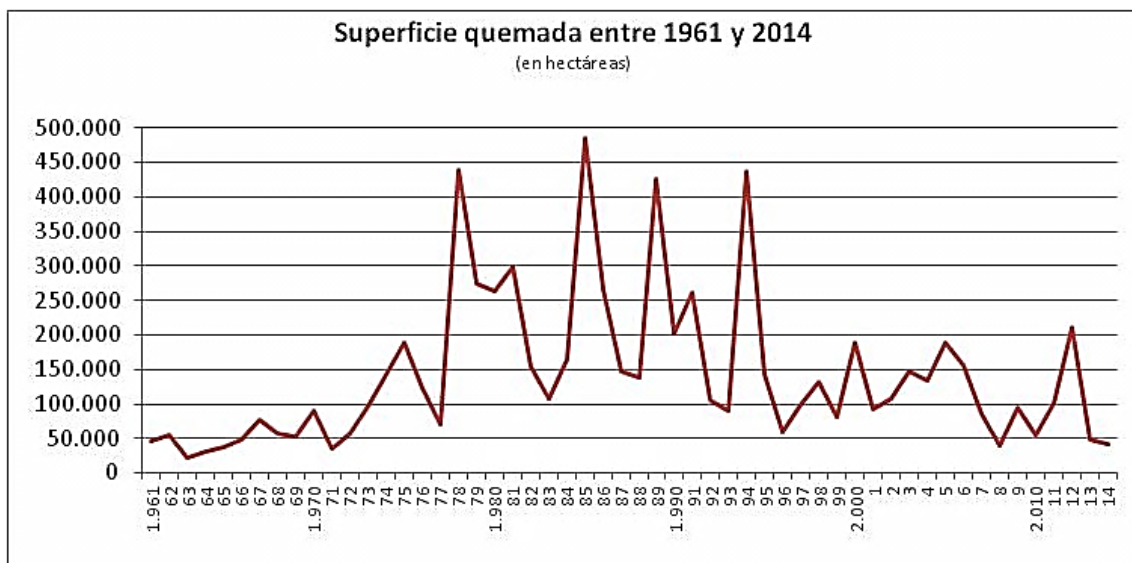


Fig 1.1- Superficie quemada en España [2]

En los últimos años, la mayoría de los incendios peninsulares se han concentrado en el noroeste, siendo Portugal, Galicia y Asturias las zonas más afectadas, como se puede ver en la Fig 1.2. En estas zonas el clima no es especialmente cálido, por ello sorprenden estos datos, pero se pueden explicar por el gran abandono forestal, el abandono agrario y la despoblación, que provocan que una vez producido, el incendio, este pueda propagarse con rapidez.

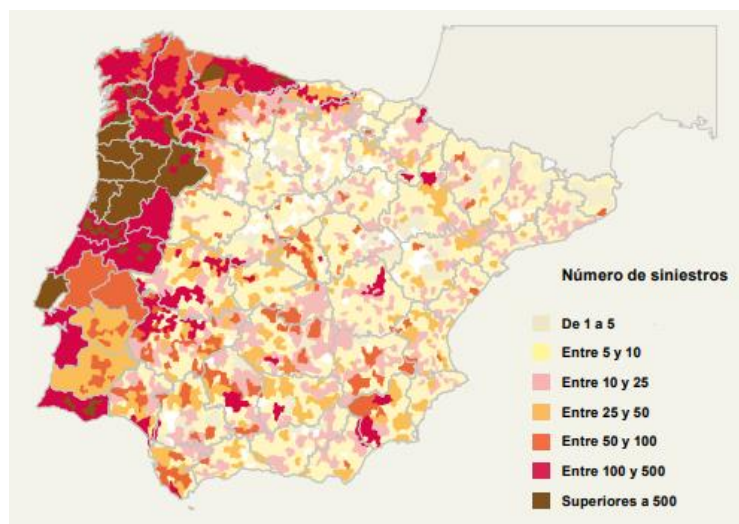


Fig 1.2- Incendios por término municipal en la Península Ibérica (2001-2010). [1]

También se observa la importancia del calentamiento global, que tiene dos efectos. El primero es que en caso de que se produzca un incendio al haber mayor sequedad en el ambiente el incendio se propaga con mayor velocidad. El segundo es el cambio en la distribución de los incendios a lo largo del año (ver Fig 1.3), esto se debe a la disminución de las precipitaciones en primavera y otoño, por lo que se alarga la temporada de calor, y con esta la de incendios.

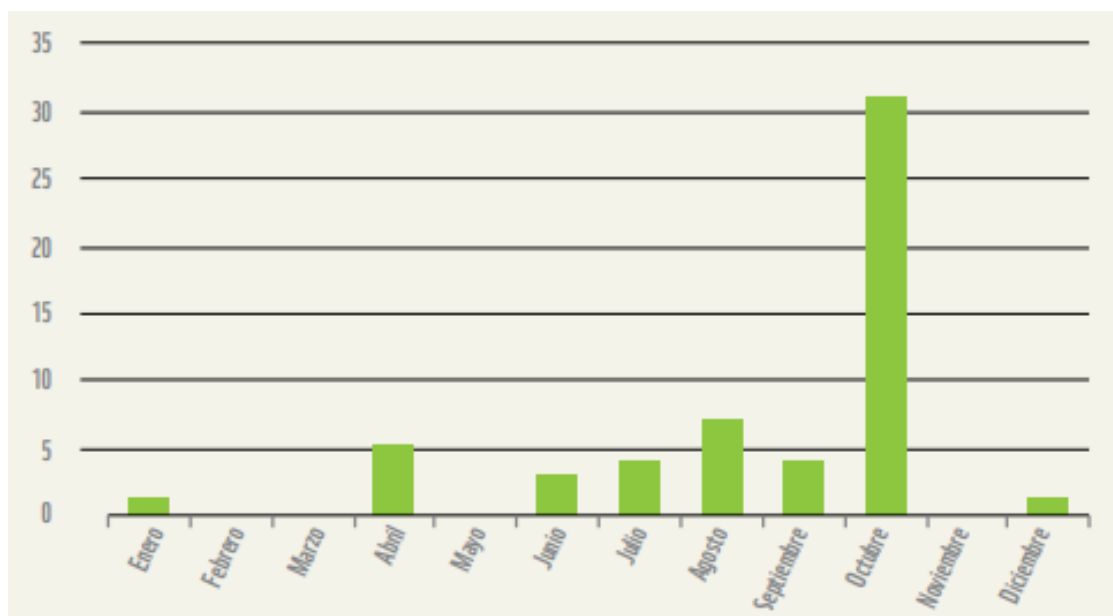


Fig 1.3- Distribución mensual de los incendios en 2017. [1]

También es importante resaltar la impredecibilidad de los incendios, dependiendo sus daños, de las condiciones que se hayan dado en los meses anteriores, haciendo prácticamente imposible realizar estimaciones sobre los daños que se van a producir o que zonas serán las más afectadas.

1.2.2 Causas de los incendios en España

Los incendios provocados por el hombre suponen un 96% del total. Las principales causas son las prácticas agrícolas y ganaderas en las que se hace uso del fuego para liberar zonas de pasto. También se debe destacar el gran número de incendios provocados por enfermos mentales (5.68%).



Fig 1.4- Principales causas de los incendios en España. [2]

Por el contrario, existen una gran cantidad de falsas teorías sobre los orígenes del fuego, entre las que podemos destacar las siguientes:

- **Terrorismo incendiario:** Los informes de la fiscalía han dictaminado que no existen pruebas de mafias criminales que actúen de manera planeada. La mayoría de incendios vienen provocados por un uso excesivo del fuego en la época de mayor inflamabilidad, tal y como indican la Policía y la Guardia Civil. Afirmar la existencia de organizaciones es una afirmación errónea que no debe simplificar un problema mucho más complejo.
- **Incendios para urbanizar:** Tras la reforma de la *Ley de Montes de 2015* se repiten noticias falsas en las redes sociales que indican que hay fines especulativos en los incendios. Sin embargo a pesar de que la reforma incluye la posibilidad de recalificar superficie quemada antes de que hayan transcurrido 30 años desde el incendio, esto solo se puede producir en casos de "razones prevalentes de interés público de primer orden", es decir, los terrenos tendrían que tener ya prevista su recalificación y es necesario el visto bueno de la Comunidad Autónoma. Desde la fecha de aprobación de la ley, no se han realizado obras públicas ni se ha recalificado el suelo en ninguna zona quemada.

- **Culpabilizar al Eucalipto:** El eucalipto es una de las especies que mejor arde. No hay correlación entre el número de eucaliptos en una región y la cantidad de incendios, sin embargo si existe entre la gestión que se hace del bosque o las condiciones climáticas de este y el número de incendios. Además, no presenta la misma inflamabilidad un eucalipto abandonado que uno que se encuentra en una producción activa. En resumen, la inflamabilidad del eucalipto no es tan importante como el mantenimiento que tengan los bosques.

1.2.3 Consecuencias de los incendios forestales

Los incendios forestales producen numerosos daños económicos, materiales, humanos y medioambientales. Por ejemplo, en 2017 murieron 4 personas en Galicia y 107 fallecieron en Portugal. Esto se produjo debido a que los incendios desbordaron a los servicios de extinción, lo que provocó que se propagasen a gran velocidad.

Los incendios también tienen un gran impacto sobre la vegetación, la fauna, los balances hidrológicos, la calidad del agua y de la atmósfera, la erosión del suelo y sus microorganismos y el paisaje.

1.2.4 Medios para evitar los incendios y sus daños

- **Disminuir la inflamabilidad del territorio:** Conocer las zonas con alto riesgo, planificar el paisaje. Determinar las zonas en la que están plantados eucaliptos y promover la plantación de especies menos incendiarias (robles, cerezos, nogales, castaños) con menor rentabilidad económica a corto plazo pero con capacidad de ser más rentables a largo plazo. También se debe revitalizar el medio rural ya que la mayoría de los incendios se producen en zonas semi-abandonadas, la población rural también debe asumir el riesgo y tomar medidas para protegerse, así como cumplir la normativa.
- **Disminuir el número de siniestros y acabar con la impunidad de los causantes:** Es importante conocer los motivos sociales por los que la gente provoca incendios para poder evitarlos, asimismo es fundamental ser capaz de identificar a los causantes, ya que actualmente solo se identifica al 9% de los causantes. Endurecer las penas también puede disuadir a los causantes, para evitar la sensación de impunidad que se tiene en la actualidad
- **Ser capaces de predecir la dirección en la que se va a extender el incendio:** El poseer un adecuado sistema de predicción puede ser decisivo a la hora de minimizar los daños de los incendios así como a combatir el incendio en los puntos con mayor actividad. También es fundamental el buen funcionamiento de estos sistemas para poder mantener a salvo a los bomberos, brigadistas y a la población en general,

1.3 Objetivo

El objetivo de este Trabajo de Fin de Grado es desarrollar un programa que permita predecir la dirección del viento mediante el uso de métodos CFD (Computational Fluid Dynamics).

Para ello se hará uso de **Matlab**, y su App **QUICKERSIM CFD Toolbox for Matlab**, el objetivo es diseñar un programa que permita realizar simulaciones sobre terrenos reales ante distintas condiciones ambientales.

También se busca comparar este método con otros que previamente existen, así como el método aquí desarrollado con vientos reales.

Las principales características de este TFG son:

1. Ser capaz de realizar simulaciones en terrenos 3D, con la complejidad que esto entraña.
2. Utilización de terrenos reales.
3. Conocer la velocidad del viento en cada punto del terreno, incluso en zonas con gran dificultad orográfica (valles, zonas realmente montañosas...).
4. Visualizar la trayectoria del viento en 3D.
5. Poder distinguir la importancia de la pendiente en la velocidad del viento y con ello en la expansión del fuego.
6. Respecto a los proyectos anteriores, la principal ventaja de este método es que nos permite determinar la velocidad del viento en cada uno de sus puntos, con una precisión muy alta, no solo su módulo, sino también su dirección y sentido. Como se puede entender, esto es un aspecto fundamental.

Por último destacar que es un método intuitivo, rápido y basado en un software muy potente como es Matlab.

1.4 Estructura

Este Trabajo de Fin de Grado presenta la estructura mencionada a continuación:

- Capítulo 2: Estudio de los modelos de simulación existentes, así como todas las herramientas utilizadas en España. El objetivo es conocer los otros programas para así poder comparar el creado, así como ver sus principales características y tratar de implementarlas en el nuestro.
- Capítulo 3: Explicación de los conceptos teóricos utilizados, así como del desarrollo de las simulaciones y los inputs necesarios.
- Capítulo 4: Muestra de resultados, tanto en mapas reales como en ficticios.
- Capítulo 5: Comparación de los datos provenientes del simulador con datos reales, así como con datos obtenidos mediante otros métodos.
- Capítulo 6: Conclusiones, análisis y mejoras. En este último se incluye un marco regulador, con un pequeño análisis de la legislación aplicable sobre el uso de programas como Matlab para la prevención de incendios forestales, y sobre el propio programa trabajado y su aplicación. También se incluye un análisis socio-económico del proyecto.

2. Estado del arte

Ha quedado demostrada la importancia de proporcionar a los servicios de extinción de incendios modelos que permitan pronosticar la dirección en la que va a avanzar el fuego, para poder anticiparse a este y extinguirlo lo más rápido posible.

Los programas deben ser capaces de determinar la trayectoria que va a recorrer el fuego. Predecir trayectorias es un fenómeno realmente complejo, por lo que es importante que se trate desde distintas aproximaciones, al depender de distintos factores. Como es evidente la sencillez de los modelos proporciona mayor velocidad de cálculo.

En los siguientes apartados se estudian los distintos modelos y programas que se pueden encontrar en el mercado, así como sus puntos fuertes, sus debilidades y aquello que les diferencia del resto.

2.1 Modelos físicos (ver [3] y [4])

En este punto se incluyen modelos que se basan en las leyes físicas y químicas en función de sus variables. Estos se basan en leyes de combustión y de transferencia de calor, dando lugar a soluciones numéricas en función del tiempo y del espacio. También están incluidos en esta categoría modelos basados en dinámica de fluidos.

Dentro de esta categoría se pueden encontrar 3 tipos de variables o parámetros físicos:

- **Terreno:** Las características del terreno son fundamentales. Entre estas se encuentran la pendiente (con su efecto sobre la trayectoria), la rugosidad y la textura, que condicionarán el avance de las llamas.
- **Combustible:** En este punto se incluyen las características que permiten que un terreno arda más o menos fácilmente y se estudian las especies que conforman la vegetación, así como la frondosidad de estas. Por su parte, es en esta categoría donde la acción del hombre es fundamental para evitar la propagación de incendios forestales, ya que la prevención de un incendio se basa en reducir la inflamabilidad del terreno mediante la planificación territorial y desarrollo rural.
- **Climatología:** En ella se pueden encontrar distintos subfactores:
 1. La radiación solar, que disminuye la humedad del bosque, lo que aumenta su inflamabilidad y combustibilidad.
 2. La temperatura, un factor fundamental a la hora de evaluar el riesgo de un incendio. Cuando las temperaturas son altas disminuye la energía calórica necesaria para la ignición, y aumenta la velocidad de propagación del incendio.

3. Humedad relativa: Esta afecta de formas distintas a la extensión de un incendio. Una mayor humedad relativa implica menor cantidad de oxígeno en el aire, por lo que este está menos disponible para la combustión. Un aumento de la humedad relativa también aumenta la humedad de los combustibles, haciendo necesaria mayor energía en el precalentamiento. La humedad relativa depende de la hora del día, alcanzando sus valores mínimos después del mediodía, creando a estas horas unas condiciones muy propicias para la extensión de los incendios forestales.
4. Viento: Es una variable fundamental, ya que acelera el proceso de combustión, provoca focos secundarios al trasladar material a zonas no afectadas y puede disminuir la seguridad de los bomberos (un cambio inesperado de dirección puede amenazar seriamente su seguridad al dejarles atrapados).

2.2 Modelos empíricos (ver [5] y [6])

Son los modelos basados en el tratamiento estadístico de datos tomados en incendios y quemas experimentales. En este tipo de modelos no se incorpora ninguna relación entre las variables físicas y las variables a estudiar sino que tratan de medir la importancia de las variables anteriormente estudiadas mediante mero tratamiento estadístico. Estos programas son útiles debido a que son mucho más simples que los programas que estudian las variables físicas. Se puede dividir en dos clases en función de sus experimentos:

- **Experimentos en terreno real:** Se realizan fuegos controlados y se estudia cómo se propagan en función de ciertas variables.
- **En el laboratorio:** Se trata de recrear en el laboratorio situaciones similares a las que se dan en los incendios. En este tipo de experimentos se tiene un mayor control de las variables, se pueden realizar un mayor número de ensayos y tomar un mayor número de datos. Sin embargo al tratarse de ensayos realizados en laboratorio la fiabilidad es menor y por tanto es necesario contrastar los datos obtenidos. Resultan sustancialmente más baratos y seguros que los anteriores.

El coste en este tipo de modelos no se produce a la hora de aplicar el modelo, sino que se produce a la hora de recoger los datos, siendo estos más caros de aplicar que los modelos físicos. El sistema canadiense y el australiano de predicción de incendios utilizan este tipo de modelos, que son especialmente eficaces en los ecosistemas en los que se ha desarrollado. Por ejemplo, en España el Centro de Investigaciones Forestales de Lourizán ha desarrollado distintos modelos empíricos a partir de datos obtenidos de quemas experimentales en masas de matorral de Galicia [7].

2.3 Modelos de Simulación Computacional de Incendios (MSCI) (ver [5] y [8])

Los modelos computacionales más empleados son los MSCI, programas informáticos que determinan los valores de determinadas variables y parámetros. Lo realizan a partir del cálculo de ecuaciones diferenciales y algebraicas. Estos modelos pueden clasificarse en dos categorías:

- **Modelos de Zona:** La principal característica de los modelos de zona es que el recinto se encuentra dividido en dos regiones. La región superior está constituida por los gases calientes que provienen de la combustión, y la inferior contiene el aire fresco. Se considera que la temperatura o concentración de gas son constantes en cada una de las regiones y a medida que se desarrolla el incendio la superficie entre ambas regiones se desplaza.

Las simplificaciones que se producen en estos modelos son más sustanciales que en los modelos de campo, por lo que sus resultados son aproximados. Por ello es importante conocer si las predicciones se acercan a la realidad y qué condiciones son necesarias para conseguir unos resultados aceptables.

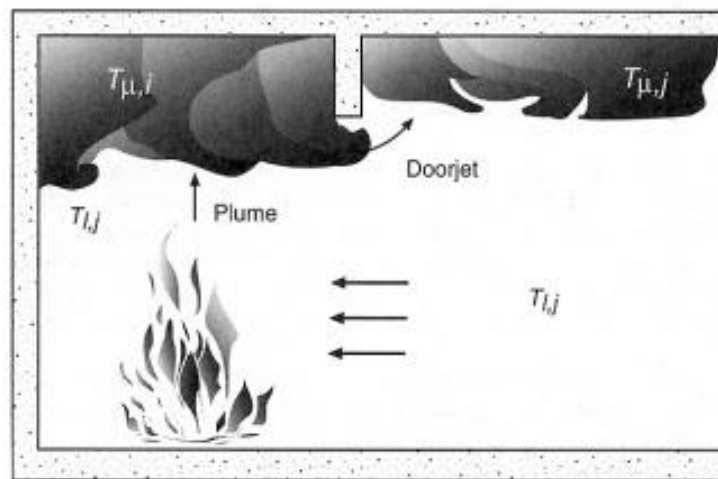


Fig 2.1- Ejemplo de modelo de zona. [8]

- **Modelos de campo:** Se basan en dividir los recintos a analizar en pequeños volúmenes de control, de tal forma que en cada celda se calculan todas las variables (temperatura, velocidad...). En este tipo de métodos se utilizan las ecuaciones de Navier-Stokes, ecuaciones complejas para las que son necesarias conocimientos algebraicos avanzados. En estos casos la solución tiene una gran dependencia de las condiciones iniciales y de frontera impuestas. El uso de modelos de campo para la predicción de incendios ha aumentado de manera exponencial en los últimos años gracias al desarrollo de aplicaciones CFD. La simulación de modelos de campo se realiza con dos tipos de herramientas informáticas:

1. Programas de resolución de fluidos generales como Fluent, CFX o la Toolbox de Matlab, que hasta ahora se han usado para otras aplicaciones tales como aeronáutica, automoción, etc. Estos modelos no sirven únicamente para predecir incendios, sino que son modelos de mecánica de fluidos capaces de resolver una gran cantidad de situaciones. Se utilizan con unas condiciones específicas para la predicción de incendios.

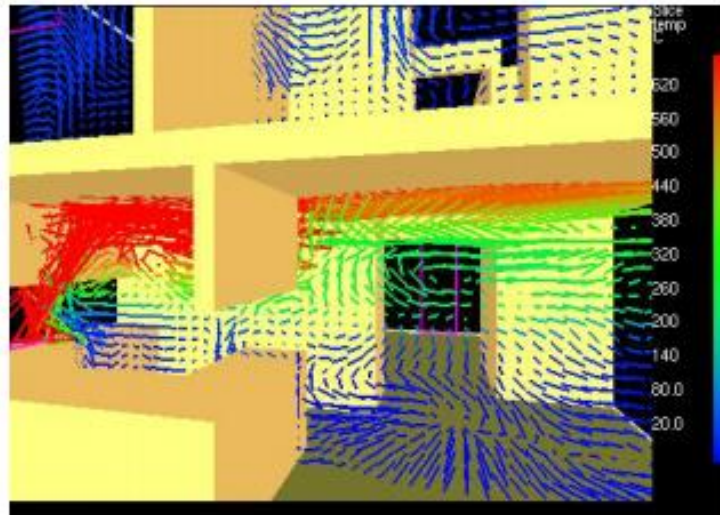


Fig 2.2- Simulación de un incendio mediante el uso de un programa CFD. [8]

2. Ciertos organismos internacionales y países han creado modelos de campo de predicción y estimación de incendios específicos como FDS, Sofie o Jasmine.

2.4 Modelos basados en el algoritmo de Fast Marching (ver [6], [9] y [10])

Los modelos basados en el algoritmo de Fast Marching parten del concepto de que la trayectoria que sigue el fuego para ir de un punto a otro es aquella que minimiza el tiempo que tarda, sin significar esto que tome el camino que minimice la distancia física.

El objetivo de los modelos de Fast Marching es determinar la trayectoria del incendio en función de las viscosidades de los fluidos. Por ejemplo, si un recipiente estuviera lleno de agua, no transmitiría igual una onda que si estuviera lleno de aceite, o de aire, al tener distintas viscosidades.

En un incendio el fuego es la onda de propagación y el campo de velocidades viene determinado por la superficie incinerada.

El método de Fast Marching trabaja con velocidades siempre son positivas, es decir, asume que el frente de onda no va a retroceder en ningún momento. En los incendios esto se cumple siempre que no intervengan factores externos como la aparición de viento o que hubiera un combustible menos inflamable de lo esperado.

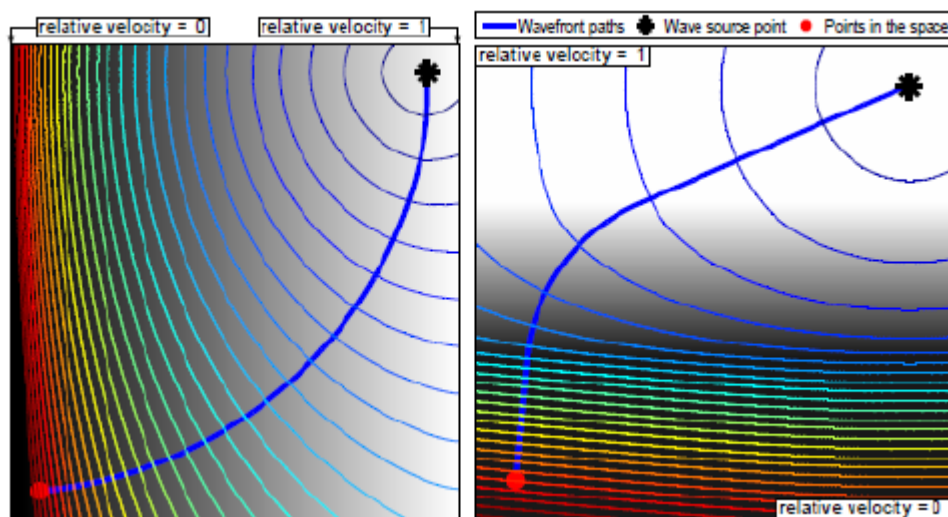


Fig 2.3-Propagación de una onda en diferentes campos de velocidades. [6]

2.5 Principales programas de predicción

2.5.1 FireRS [11]

En este caso no es un programa en sí, sino un proyecto que incluye varios programas informáticos. Fire RS es un acrónimo de wildFIRE Remote Sensing, se trata de un proyecto diseñado para detectar y gestionar incendios mediante el uso de tecnología puntera. FireRS ofrece información e manera prácticamente instantánea de la situación del fuego, mediante GPS, e imágenes infrarrojo. Este proyecto es complejo y está compuesto de sensores de tierra, un satélite dedicado, y un centro de control.

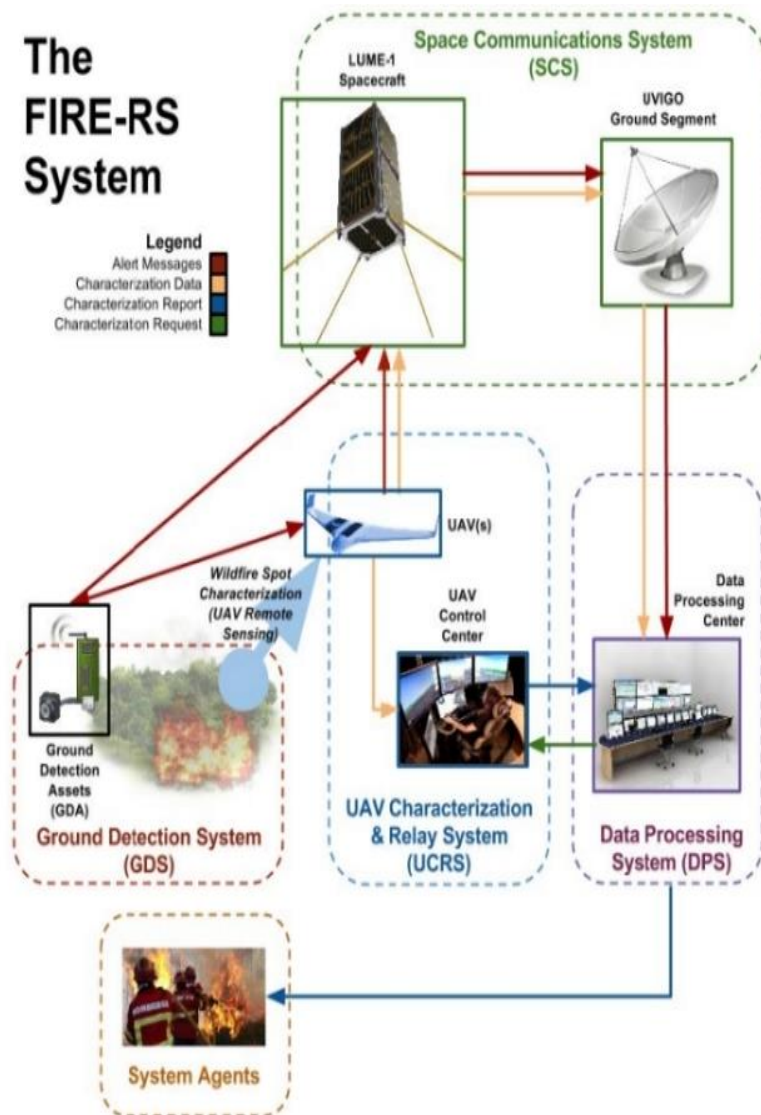


Fig 2.4- Esquema del FIRE RS. [11]

2.5.2 Prometheus [12]

Es un sistema de simulación de incendios forestales que se usa principalmente en Canadá. Es un modelo determinista, por lo que los resultados dependen directamente de la entrada.

Utiliza diferentes parámetros como la pendiente, la altura, o el clima para realizar simulaciones haciendo uso de un modelo computacional cuya base es el principio de Huygens. Este principio dice lo siguiente: “Todo punto de un frente de onda inicial puede considerarse como una fuente de ondas esféricas secundarias que se extienden en todas las direcciones con la misma velocidad, frecuencia y longitud de onda que el frente de onda del que proceden.” [13]

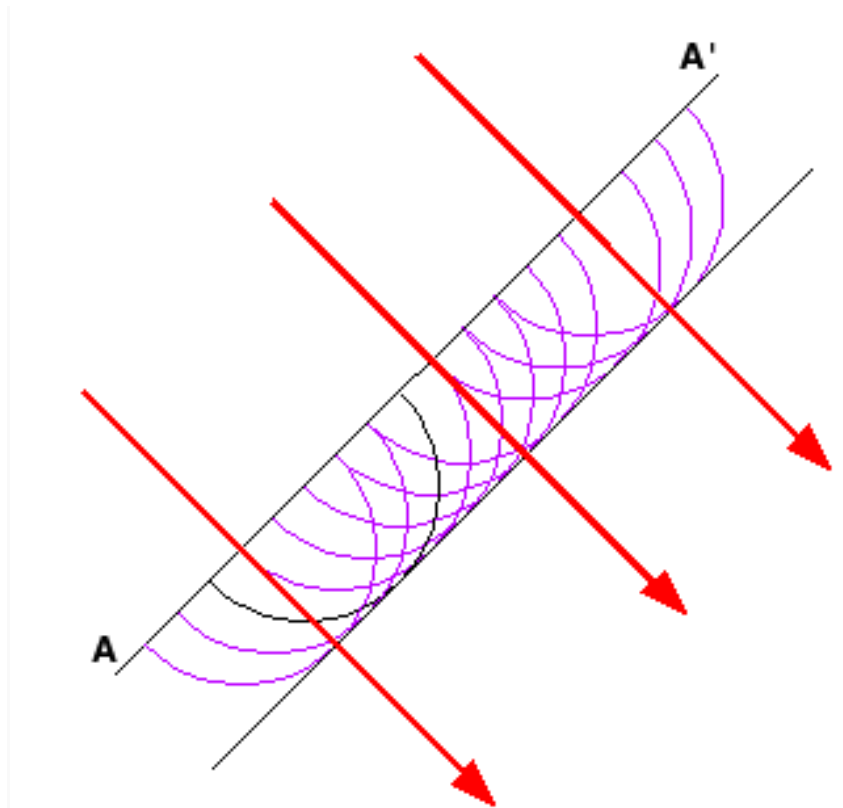


Fig 2.5- Frente de onda mediante el principio de Huygens.[13]

Para poder realizar simulaciones con este modelo, se debe asumir que el fuego es una figura geométrica cuyos extremos extienden esta onda de manera diferente en función de las características de este extremo (clima, altura...)

Para funcionar correctamente, Prometheus necesita inputs en formato ASCII con distintas informaciones tales como altura, meteorología, zonas especiales...

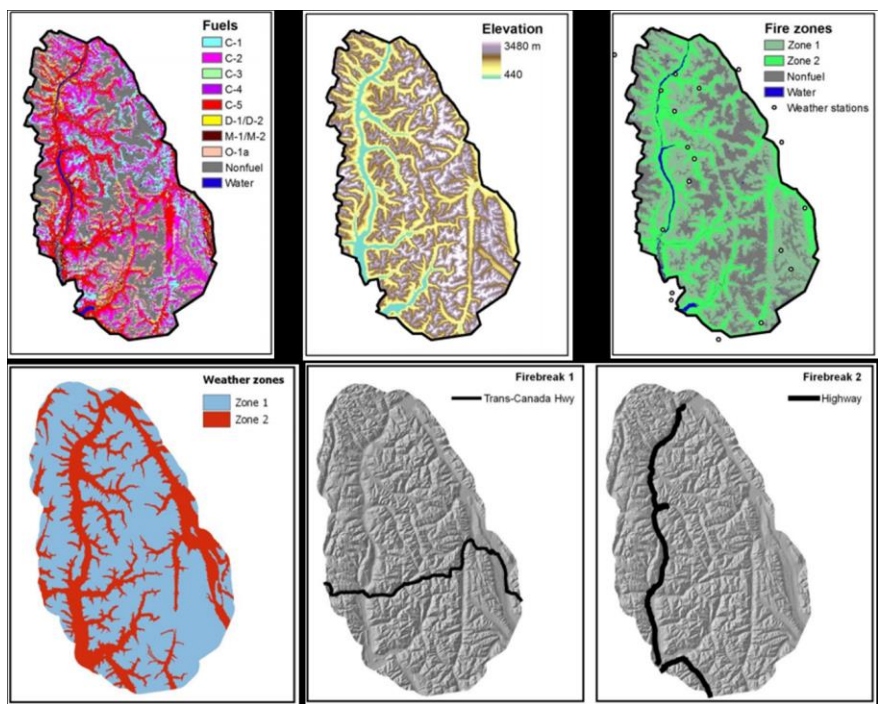


Fig 2.6- Esquema con los inputs introducidos. [12]

2.5.3 FARSITE [14] y [15]

FARSITE es un modelo determinista que usa información sobre la topografía, combustible, clima y viento. FARSITE se usa predominantemente en los servicios forestales estadounidenses, así como otras instituciones federales y agencias encargadas de prevenir los incendios. Al ser un programa realmente complejo es necesario poseer un gran conocimiento del mismo para poder utilizarlo de manera correcta.

FARSITE incorpora los siguientes modelos sobre cómo se comporta el fuego:

- Modelo de propagación de incendios superficiales de Rothermel (1972)
- Modelo de predicción de Van Wagner's (1977)
- Modelo de Rothermel's de extensión del fuego (1991)
- Modelo de localización de Albini (1979)
- Modelo basado en la humedad del combustible Nelson's (2000)



Fig 2.7- FARSITE. [15]

En la siguiente figura se observa el diagrama de flujo del programa, todas las entradas que necesita, así como todas las salidas y datos que se puede obtener de este.

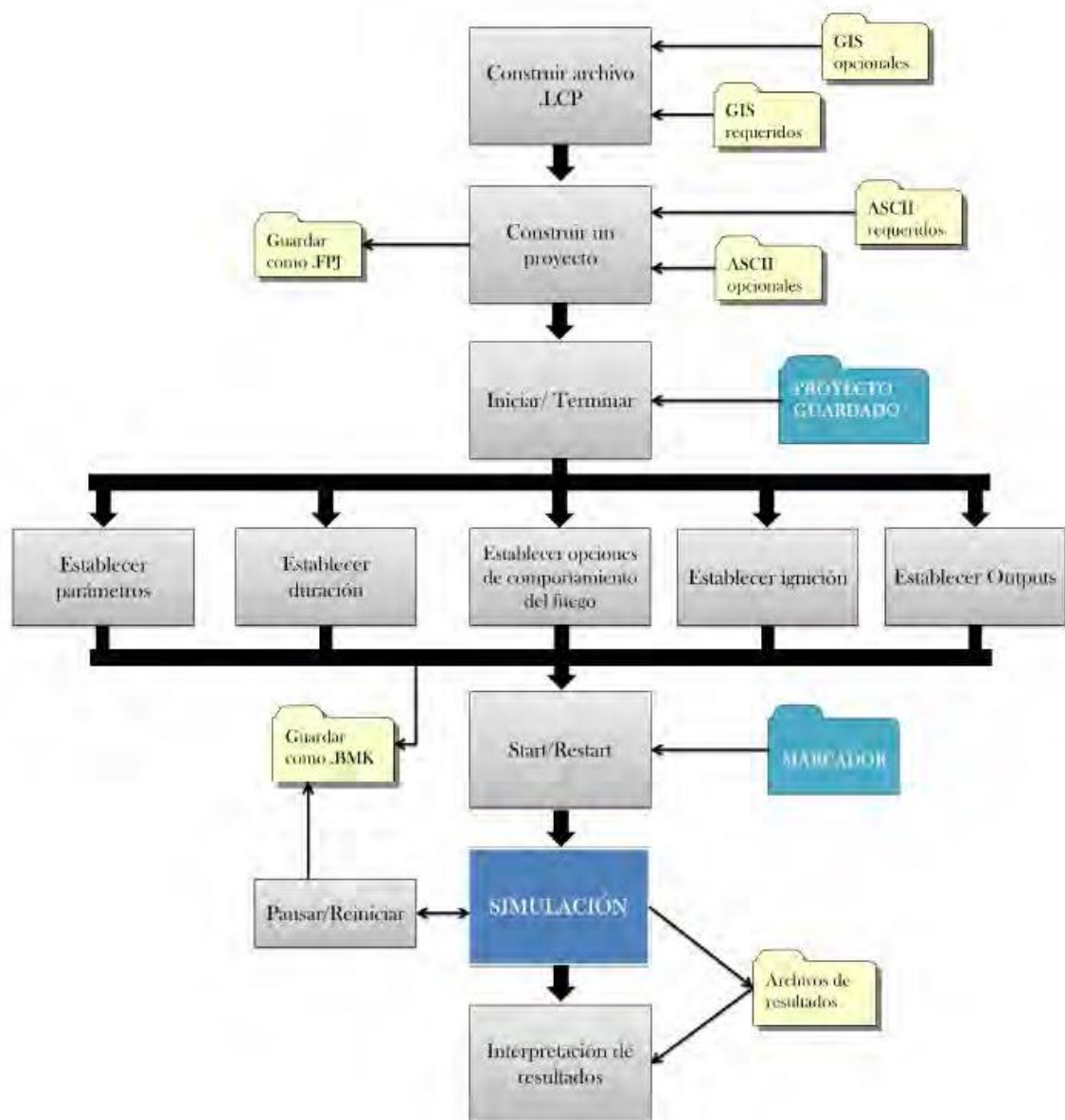


Fig 2.8- Esquema de funcionamiento de FARSITE. [15]

En la imagen que se muestra a continuación se visualizan los resultados de una simulación realizada con FARSITE. Uno de los principales inconvenientes de este método es la mala calidad que proporciona en la representación de resultados. Como se puede observar, el programa representa líneas poco nítidas, haciendo complicado distinguir el terreno del incendio.

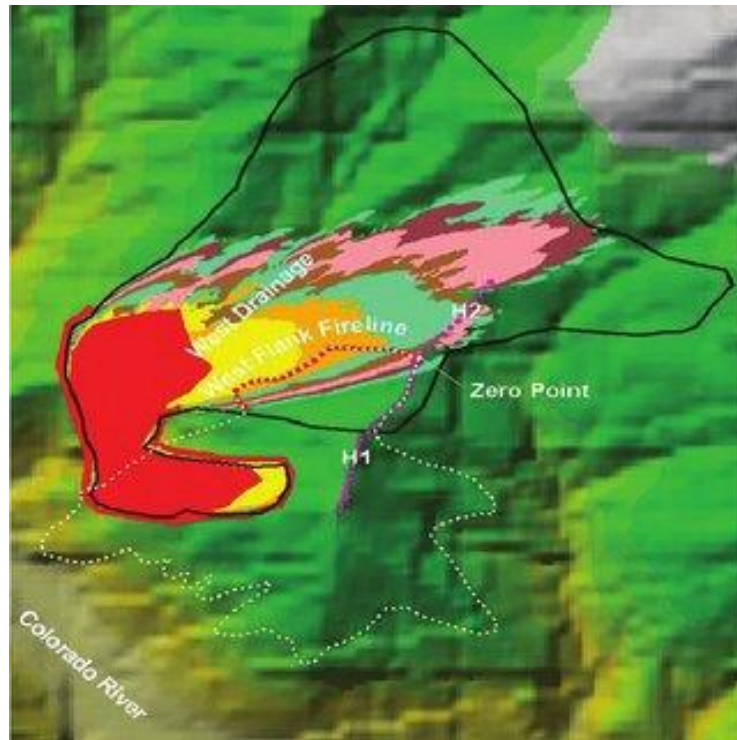


Fig 2.9- Ejemplo de simulación de Farsite. [14]

2.5.4 Pyrowarn [16]

Estudiantes de la Universidad Politécnica de Valencia han creado un programa que permite evaluar los incendios forestales. Este programa se basa fundamentalmente en datos geográficos y climáticos.

Esta herramienta no solo permite predecir la dirección del incendio una vez se ha producido, sino que evalúa los lugares con mayor riesgo en un momento determinado, cuando alguna de sus variables toma un valor muy alto (temperatura, baja humedad...)

El programa aún no tiene un uso muy extendido, sin embargo, incluye más posibilidades que los programas vistos con anterioridad.

3. Desarrollo del trabajo

El código desarrollado en este TFG permite el uso de mapas de elevación reales para realizar las simulaciones, pudiendo variar la dirección y la intensidad del viento. El objetivo es que sea sencillo para que personas sin conocimientos sobre programas CFD sean capaces de hacer uso de él. También es importante ofrecer una clara visualización de los resultados.

En los siguientes apartados se realiza en primer lugar una introducción de los conceptos teóricos utilizados, para posteriormente explicar la metodología utilizada para obtener los mapas de elevación reales y por último, mostrar un diagrama de flujo del proceso.

3.1 Conceptos teóricos (ver [17] y [18])

Como ha sido mencionado con anterioridad el objetivo de este trabajo es el desarrollo de un código de MATLAB que mediante el uso de técnicas CFD permita predecir la dirección del viento para mejorar la prevención de incendios forestales.

Se va a utilizar un programa de CFD específico denominado QuickerSim [19], un programa desarrollado por la Warsaw University of Technology. Este software permite adaptar de manera sencilla y flexible el código para realizar simulaciones de flujo y calor. En el siguiente epígrafe se pretende explicar al lector el funcionamiento de nuestro programa y los motivos por los que se ha elegido para resolver este problema.

El programa de CFD hará uso de las ecuaciones de Navier-Stokes, que describen matemáticamente, el movimiento de un fluido. Estas se obtienen aplicando los principios de conservación de la mecánica y la termodinámica a un volumen fluido y permiten determinar el movimiento futuro de un fluido a partir de su estado inicial.

Esta expresión representa el principio de conservación lineal para cualquier fluido:

$$\rho \frac{Du_i}{Dt} = \rho F_i - \frac{\partial P}{\partial x_i} + \frac{\partial [2\mu(e_{ij} - \Delta\delta_{ij}/3)]}{\partial x_j}$$

Y la ley de conservación de la masa es la siguiente:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0$$

Siendo ρ la densidad u_i ($i = 1, 2, 3$) las componentes de la velocidad, F_i el campo de aceleraciones, P la presión del fluido y μ la viscosidad dinámica.

$$e_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$$

$\Delta = e_{ij}$ es la divergencia del fluido y δ_{ij} la delta de Kronecker. $\frac{D}{Dt}$ es la derivada total, siguiendo el fluido:

$$\frac{D}{Dt}(\cdot) \equiv \frac{\partial(\cdot)}{\partial t} + (V \nabla)(\cdot)$$

Las ecuaciones no son lineales debido al término relacionado con la derivada total, cuando μ es constante las ecuaciones se simplifican de la siguiente manera (expresadas en forma vectorial):

$$\rho \frac{Du}{Dt} = \rho K - \nabla P + \mu \left(\frac{1}{3} \nabla(\nabla u) + \nabla^2 u \right)$$

Estas ecuaciones son las que se utilizan para resolver el problema, para más información sobre el desarrollo de las ecuaciones de Navier-Stokes se remite al trabajo de Jason M. Forthofer [20], en el que se analiza cada término de manera independiente y su posterior desarrollo.

3.2 Mapas de elevación del terreno

Para que el programa funcione, es necesario introducir un fichero de datos .mat que contenga la matriz de elevación del terreno. Esta matriz debe de ser cuadrada $a \times a$, y puede tener cualquier tamaño aunque es recomendable que no sea demasiado grande para que el programa funcione adecuadamente.

La matriz que se importa contiene en cada posición(x (columna), y (fila)) la altura de la matriz en ese punto, es decir, el valor z.

3.2.1 Prueba con un mapa de elevación ficticio

Se puede usar cualquier mapa de elevación para realizar nuestras pruebas, en este caso se ha utilizado la famosa función de Matlab *Peaks*, que se ha usado en valor absoluto y escalado, multiplicando sus valores por un número mayor, para aumentar la pendiente de esta.

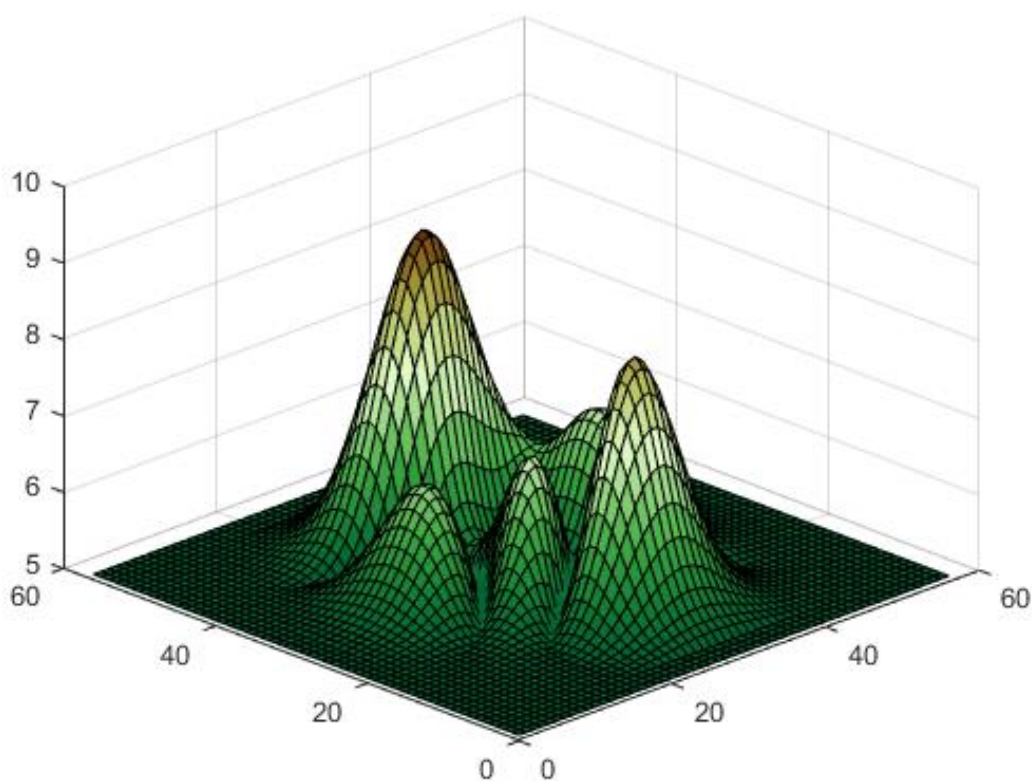


Fig 3.1- Ejemplo de mapa de elevación ficticio, Peaks

3.2.2 Prueba con un mapa de elevación real

Usar mapas de elevación ficticios facilita la realización de pruebas, pero es necesario que la aplicación tenga un comportamiento adecuado en mapas de elevación reales, porque presentan formas más complejas que los anteriores. Lo verdaderamente interesante de la aplicación es que sea capaz de simular la propagación de un incendio en un terreno real.

En España no hay bases de datos con los mapas de elevación del terreno suficientemente completas y accesibles (hay cierta información en la web del ministerio de fomento). Por

ello se utiliza [21] aquí hay información de los mapas de elevación de prácticamente cualquier lugar de Estados Unidos.

Los pasos para obtener los mapas son los siguientes.

- 1 Entrar en la página y seleccionar el estado sobre del que se desean los mapas.

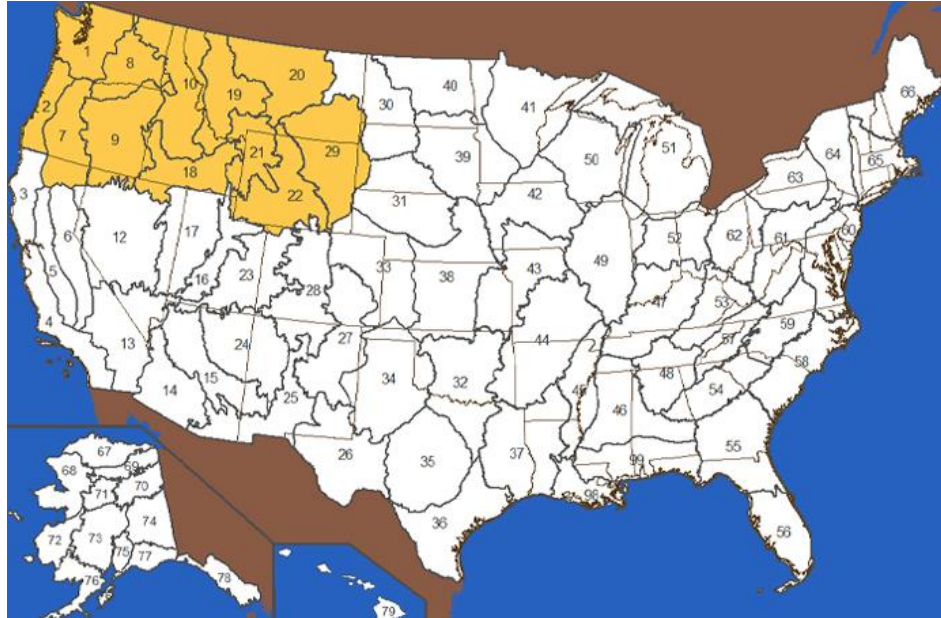


Fig 3.2- Selección del Estado [21]

- 2 Seleccionar la opción de descargar datos



Fig- 1 Selección de la opción de descarga [21]

- Elegir la superficie que se desea descargar, así como seleccionar que se desean descargar datos topográficos.

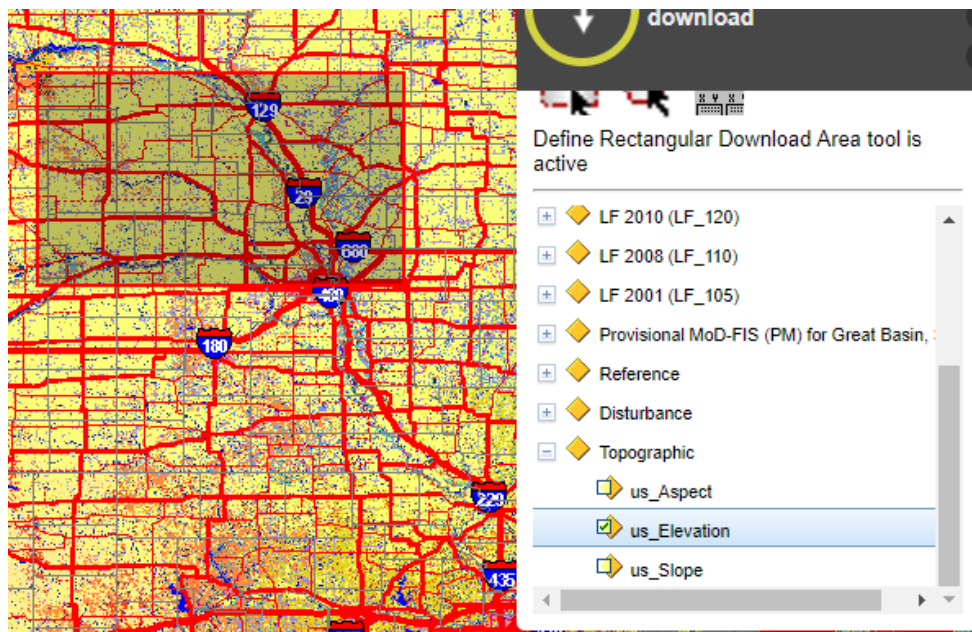


Fig 3.3- Selección de los datos topográficos [21]

- Es necesario modificar los datos para descargar, para obtenerlos en el formato deseado (TIFF).



Fig 3.4- Modificación del formato [21]

- Se selecciona como se desea descargar los mapas.

<input type="checkbox"/> us_Aspect	Not selected.
<input checked="" type="checkbox"/> us_Elevation	GeoTIFF ▼
<input type="checkbox"/> us_Slope	Not selected.

Fig 3. 5- Selección del formato [21]

- Descarga de los datos y guardarlos en una carpeta determinada.

Una vez descargados el los datos el proceso para introducirlos en Matlab es realmente sencillo, consiste en:

- Leer el formato .tiff haciendo uso de la función imread.
- Redimensionar la matriz, tanto la longitud de esta como los valores que toma.
- Una vez hecho esto ya es posible representar los mapas

```

1 - clear
2 - clc
3 - a=imread('US_DEM2016.tif');
4 - B = imresize(a,1/10);
5 - B=B/10;
6 - figure(1);
7 - surf(B);
8 - demcmap(B,10000);
9 - figure(2);
10 - surf(a)
11 - [a,b]=size(B);

```

Fig 3. 6- Código de importación de un mapa

3.2.3 Muestra de mapas reales

En las siguientes imágenes se muestran mapas reales de elevación 3D. Se han obtenido de la página de Landfire y en cada uno de ellos se señalan sus coordenadas.

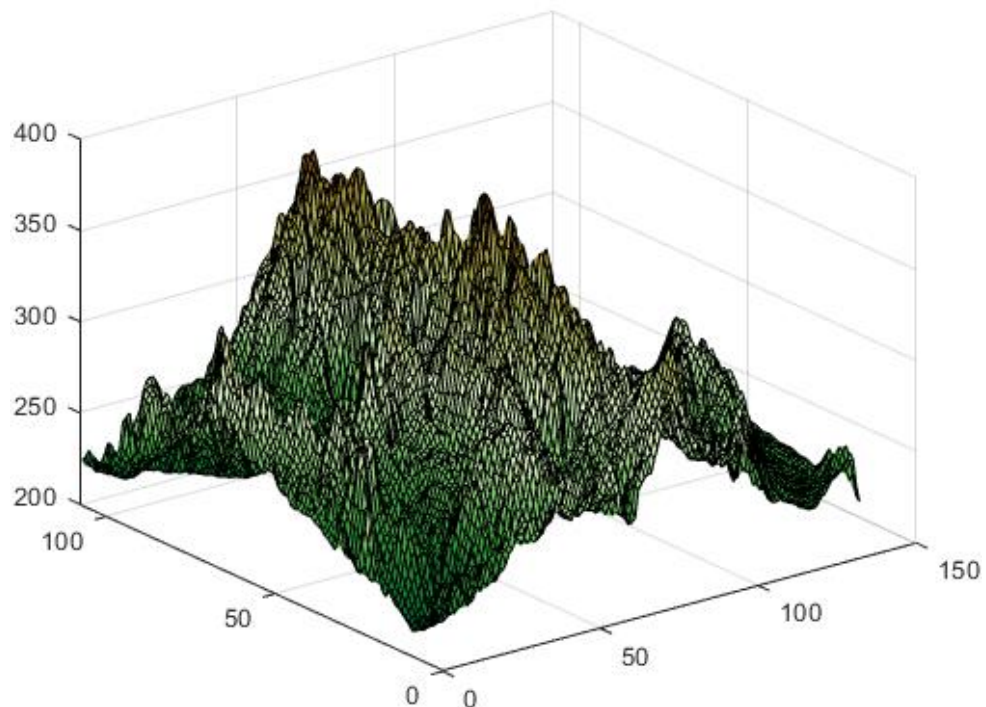


Fig 3.7- Coordenadas: N 30.59073 y W -99.37331

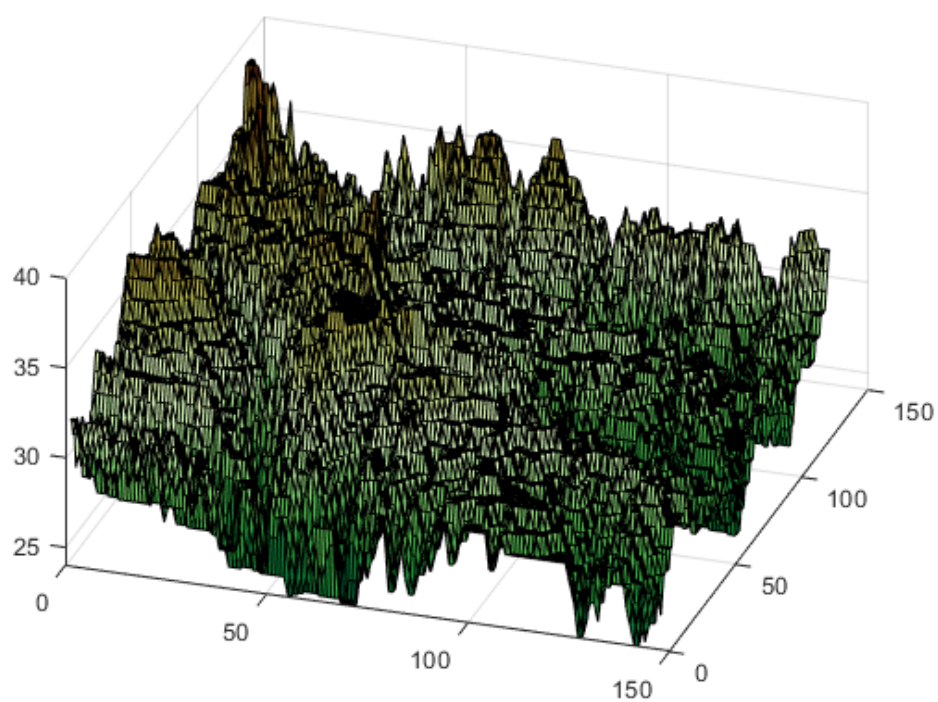


Fig 3.8- Coordenadas: N 44.43274 y W -102.6587

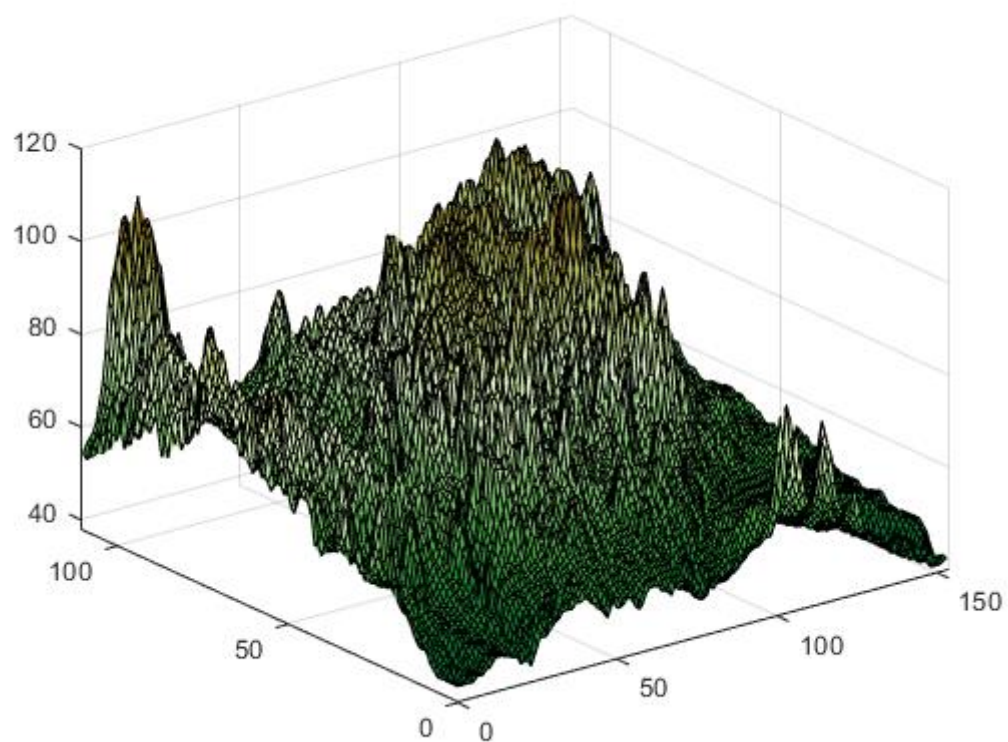


Fig 3.9- Coordenadas: N 47.28841 y W -111.51761

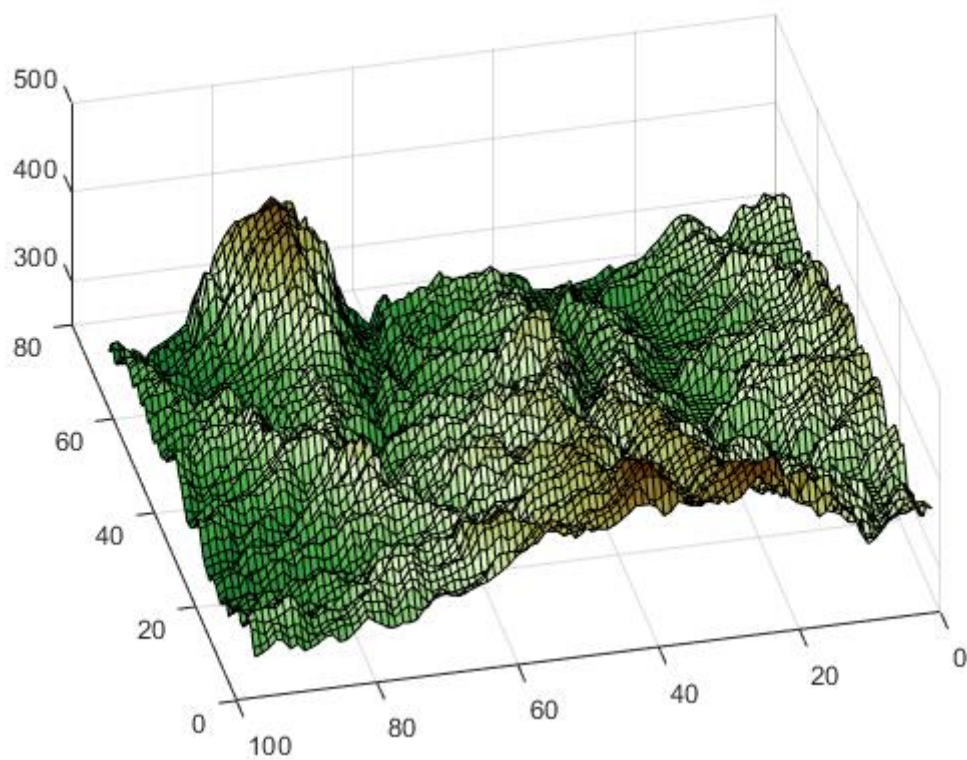


Fig 3.10- Coordenadas: S 34.24644 y E -97.25181

A continuación se muestra un diagrama de flujo con los pasos que sigue el programa para obtener, representar y analizar los resultados. En los Anexos: Código se puede encontrar el código, si se deseara analizar este de manera detallada.

Diagrama de flujo del proceso

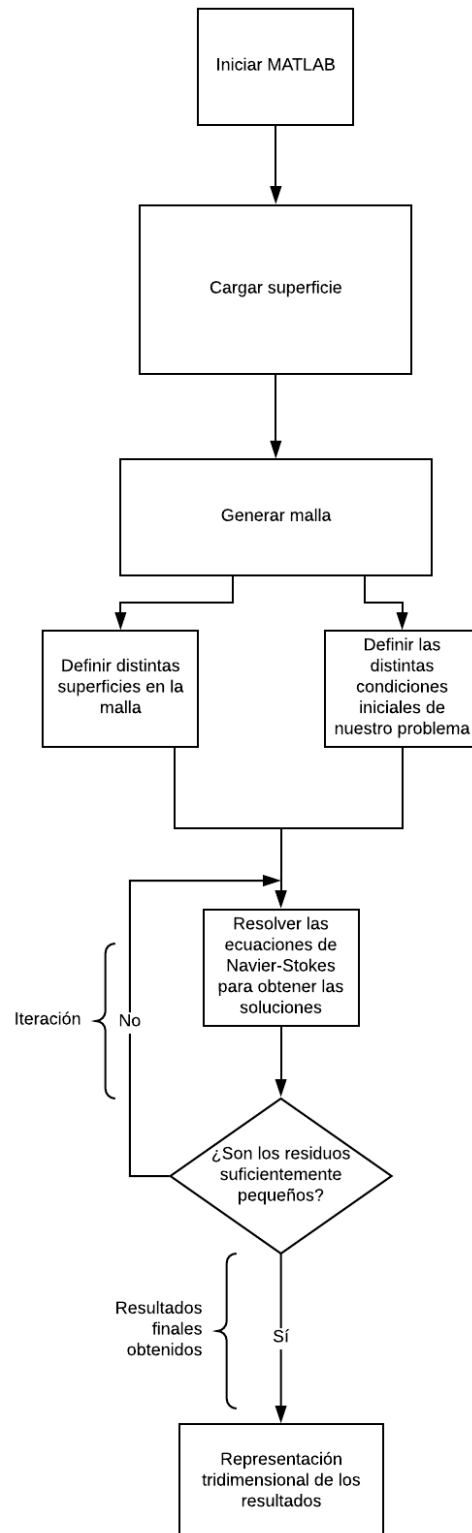


Fig 3.11- Diagrama de flujo del proceso

4. Muestra de resultados

En este apartado, se va a visualizar la trayectoria que sigue el viento, tanto en montañas reales como en montañas ficticias. Ahora bien, en ocasiones los gráficos no son demasiado buenos. Esto se debe a la gran complejidad que presenta representar trayectorias reales, y también a los problemas que tiene Matlab para representar determinados gráficos.

Las figuras reales que se han utilizado son las mostradas anteriormente, también se ha utilizado la función Peaks, que es la primera que se muestra.

4.1 Muestra de trayectorias en la superficie generada por Peaks

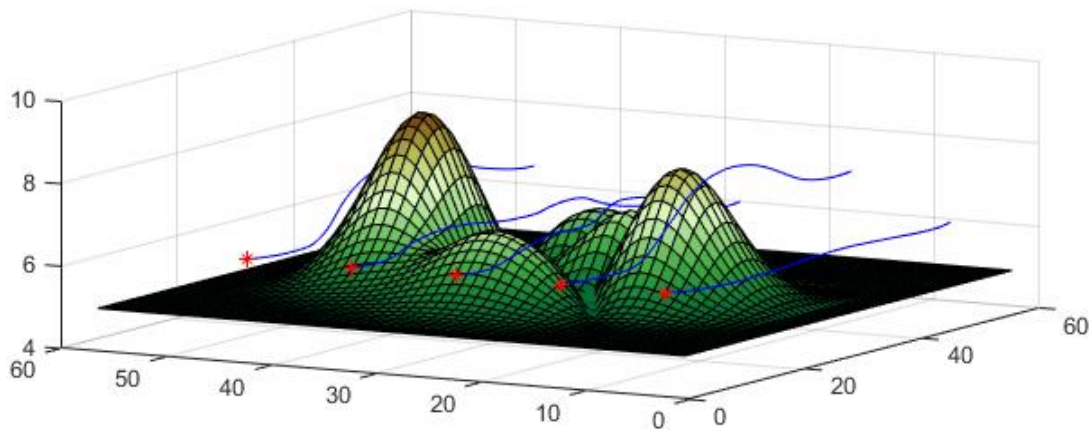


Fig 4.1- Visualización de la trayectoria del viento utilizando Peaks

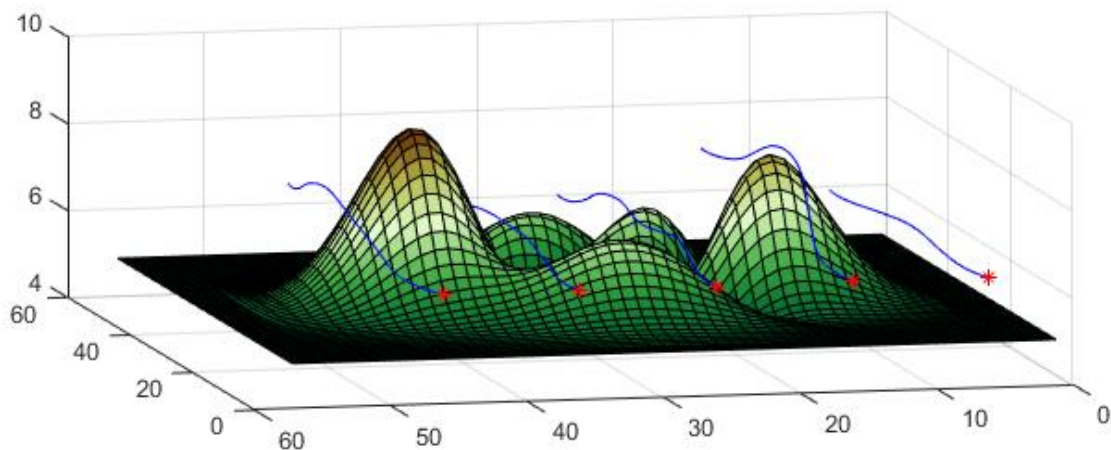


Fig 4.2- Visualización de la trayectoria del viento utilizando Peaks (2)

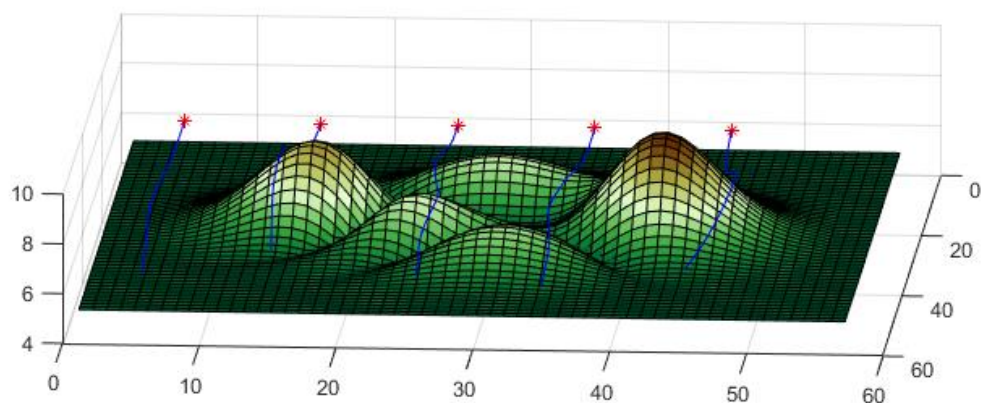


Fig 4.3- Visualización de la trayectoria del viento utilizando Peaks (3)

En la superficie Peaks se visualiza perfectamente la importancia de las montañas en la trayectoria, tal y como se muestra en las imágenes anteriores.

4.2 Muestra de trayectorias en terrenos reales

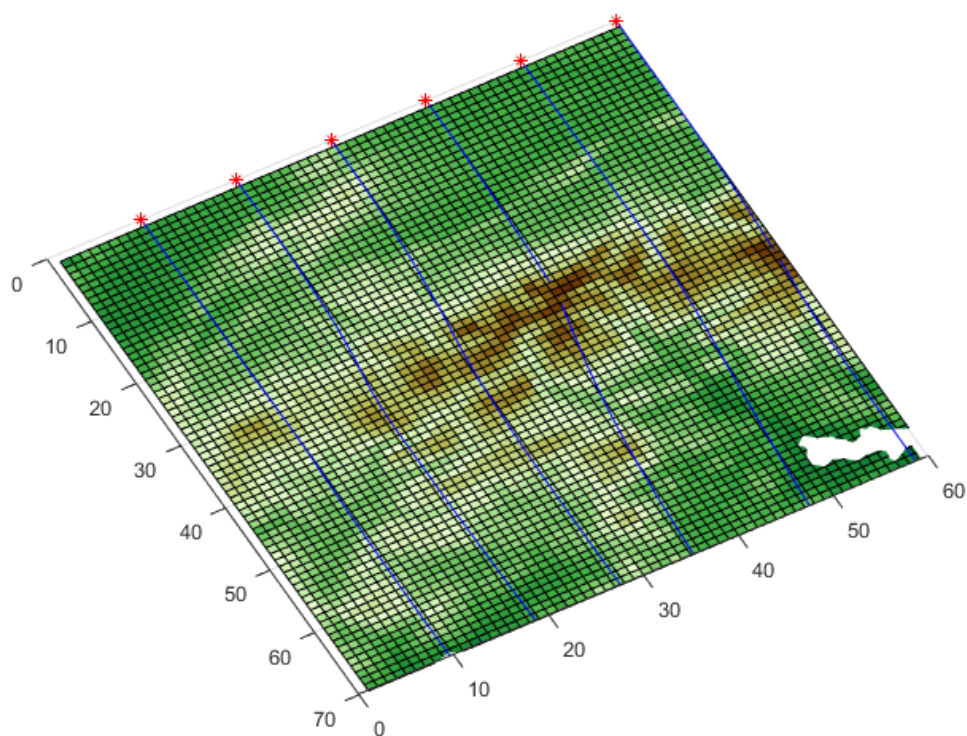


Fig 4.4- Coordenadas: N 30.59073 y W -99.37331

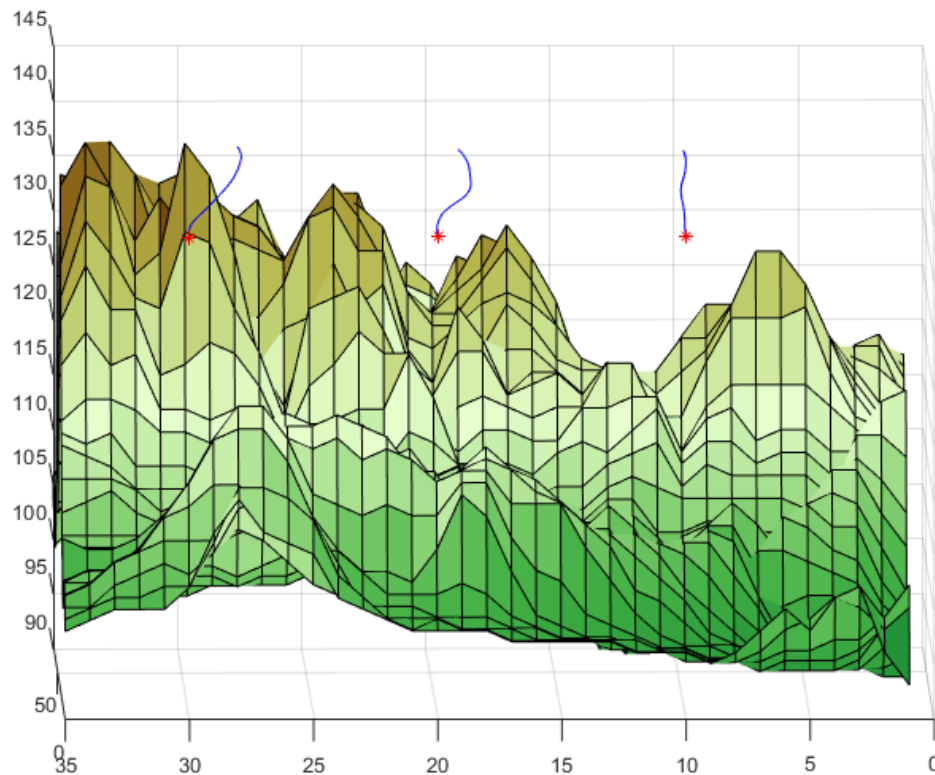


Fig 4.5- Coordenadas: N 30.59073 y W -99.37331

En esta montaña se puede visualizar la trayectoria. Como el viento evita chocar contra la montaña.

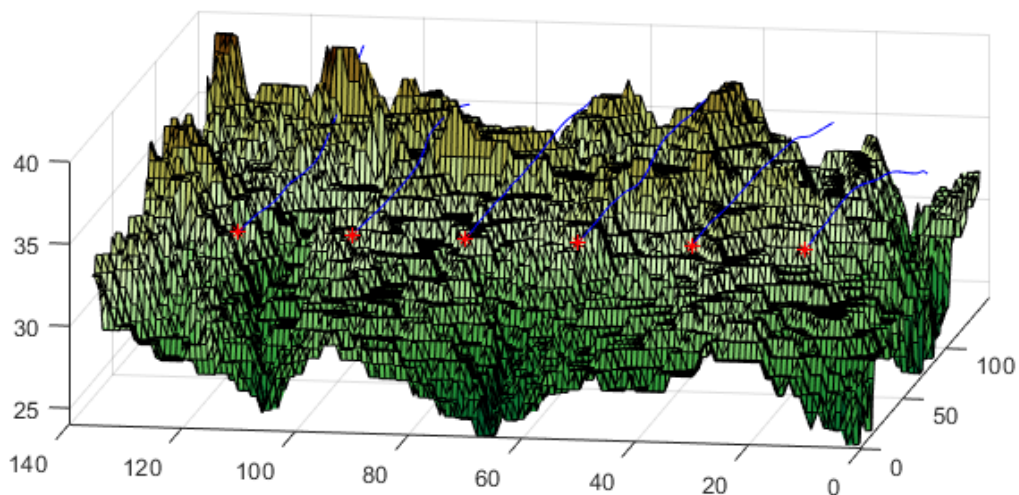


Fig 4.6- Coordenadas: N 44.43274 y W -102.6587

En esta figura se observa perfectamente la evolución de las distintas trayectorias, a pesar de que el paisaje es muy rugoso, lo que dificulta una mejor visualización.

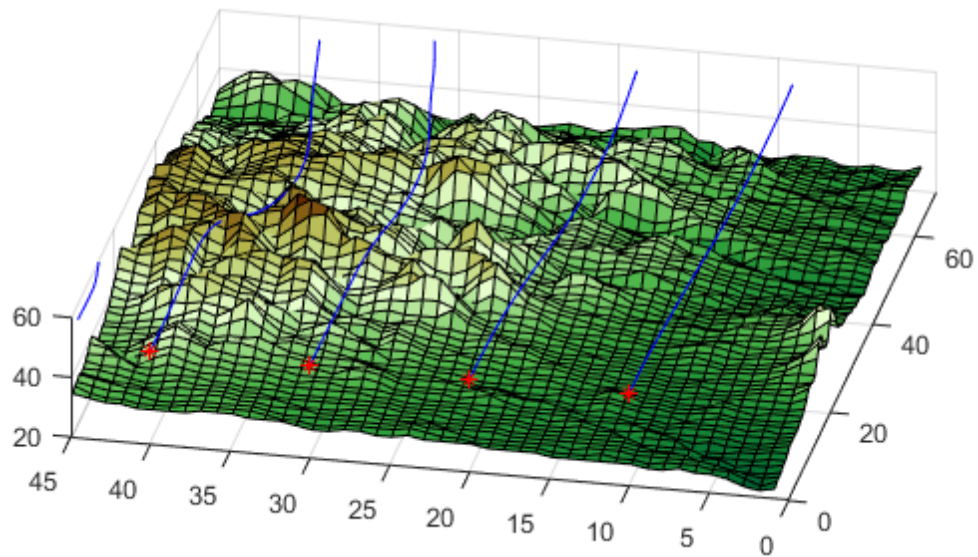


Fig 4.7- Coordenadas: N 47.28841 y W -111.51761

En esta ocasión se representa la vista superior de la montaña: se visualiza perfectamente como la orografía influye en la trayectoria del viento. En la zona menos montañosa, el viento sigue la trayectoria inicial, mientras que en la zona con mayor cambio de pendiente, ve modificada sus trayectorias.

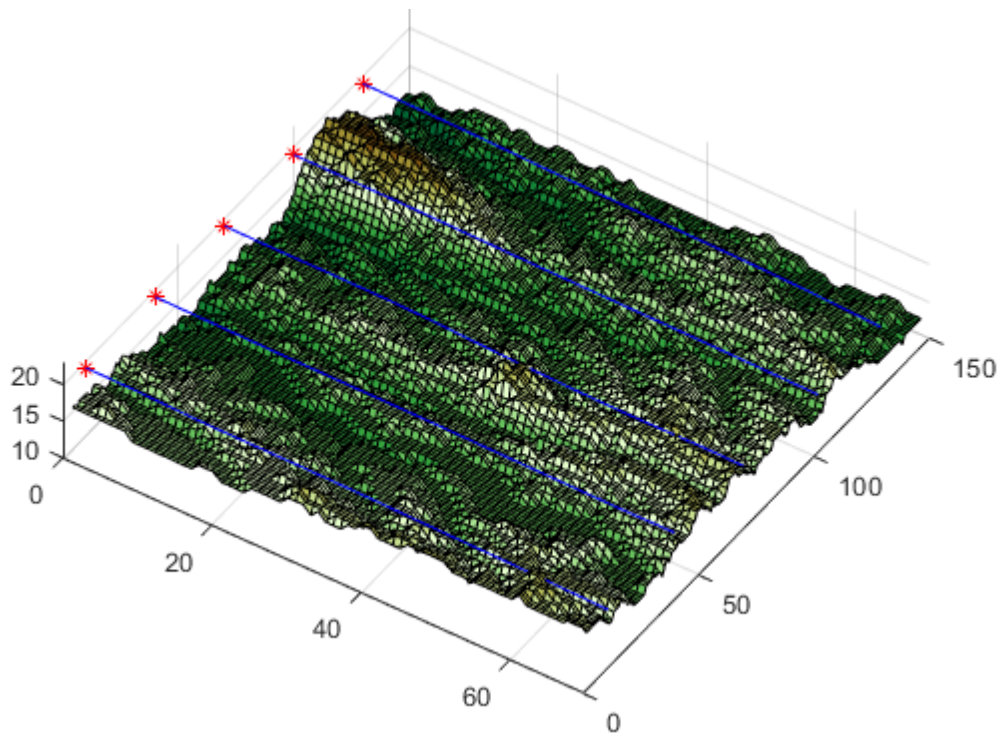


Fig 4.8- Coordenadas S 34.24644 y E -97.25181

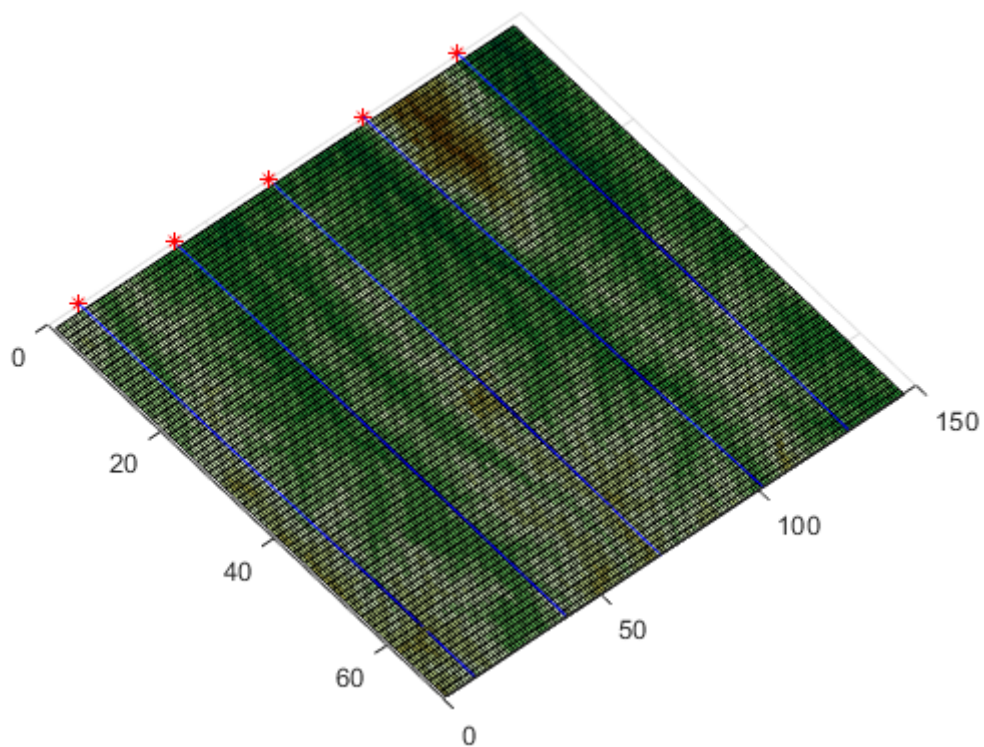


Fig 4.9- Coordenadas: S 34.24644 y E -97.25181

Se debe destacar las dificultades que tiene el visualizador de Matlab para representar estas trayectorias. El programa funciona en todos los paisajes, sin embargo, una representación correcta no es posible en todos ellos.

5. Validación del modelo

En este tipo de modelos la solución depende enormemente del diseño inicial realizado y de las condiciones de contorno (iniciales) establecidas. Por ello es fundamental validar el modelo, ya que el hecho de que un modelo funcione con unas características determinadas no significa que vaya hacerlo para otras.

5.1 Tesis de Jason M. Forthofer [20]

Para validar el modelo, se van a comparar los datos obtenidos con nuestro modelo con los datos publicados en la tesis doctoral de Jason M. Forthofer.

En efecto, su tesis trata de prever la dirección del viento y compara la predicción con unos datos reales. Para ello hace uso del programa de CFD Fluent, que es un programa de CFD general, es decir, que en función de las características detalladas resuelve distintos modelos.

5.1.1 Condiciones de contorno

Considera el aire un fluido viscoso, incompresible y turbulento. También considera que la densidad es constante.

Forthofer diseña su modelo asumiendo que hay una atmósfera estable, esto no es lo más común, pero suelen ser prácticamente estables, por lo que se puede realizar esta aproximación. A altas velocidades, como suele ser el caso de los incendios, hay suficiente mezcla en la capa límite como para ignorar los efectos de la temperatura.

Forthofer considera que los efectos que tiene la geometría del terreno sobre nuestro problema son mucho mayores a los efectos de Coriolis, por lo que estos se pueden despreciar.

5.1.2 Mallado

Las ecuaciones diferenciales están resueltas en un mallado desestructurado. Esto permite a Forthofer hacer uso de técnicas de mallado automático, y ordenar de forma más sencilla las rejillas que en un mallado estructurado. Usa un mallado hexagonal, concentrando un mayor número de celdas en los puntos más cercanos al suelo, aquellos donde se producen las mayores diferencias de velocidad.

Forthofer realiza una simulación con un mallado más fino y otra simulación con un mallado más grueso, para comprobar si el tamaño de las celdas afecta significativamente a los resultados obtenidos.

5.1.3 Askervein hill

Askervein Hill es una montaña situada en Escocia. Esta montaña ha sido objeto de un programa de recolección de datos entre 1982 y 1983 por parte de Taylor y Teunissen.

El objetivo de este proyecto era recoger datos de una atmósfera neutral en una colina para validar los modelos de predicción de vientos. Askervein Hill tiene 116 metros de altura y

está rodeada de zonas llanas, además la vegetación de esta montaña es bastante baja. Estas características hacen de Askervein Hill una montaña ideal para probar los modelos.

Se instalaron cerca de 50 torres capaces de medir la velocidad del viento de la forma más exacta posible. Las torres se encontraban separadas por 100 metros aproximadamente. Se formaron tres líneas distintas, como se puede observar en la figura que se encuentra a continuación (Fig 5.1). Los datos que se han usado son vientos medios medidos durante 10 minutos a una altura de 10 metros por encima del suelo. Específicamente los datos con los que trabaja Forthofer son de una velocidad entrante de 8.9 m/s, con una atmósfera prácticamente neutra.

En las siguientes figuras se puede comparar los resultados que obtuvo Forthofer con los reales, y más adelante son comprobados con los propios. Como se puede observar la velocidad aumenta en la parte más alta de la montaña y disminuye en la parte posterior.

Algunos autores consideran que en ciertas zonas, la velocidad de viento debería tener en cuenta a colinas cercanas no representadas para obtener mayor calidad en la predicción. Esta circunstancia podría explicar ciertos errores en las simulaciones.

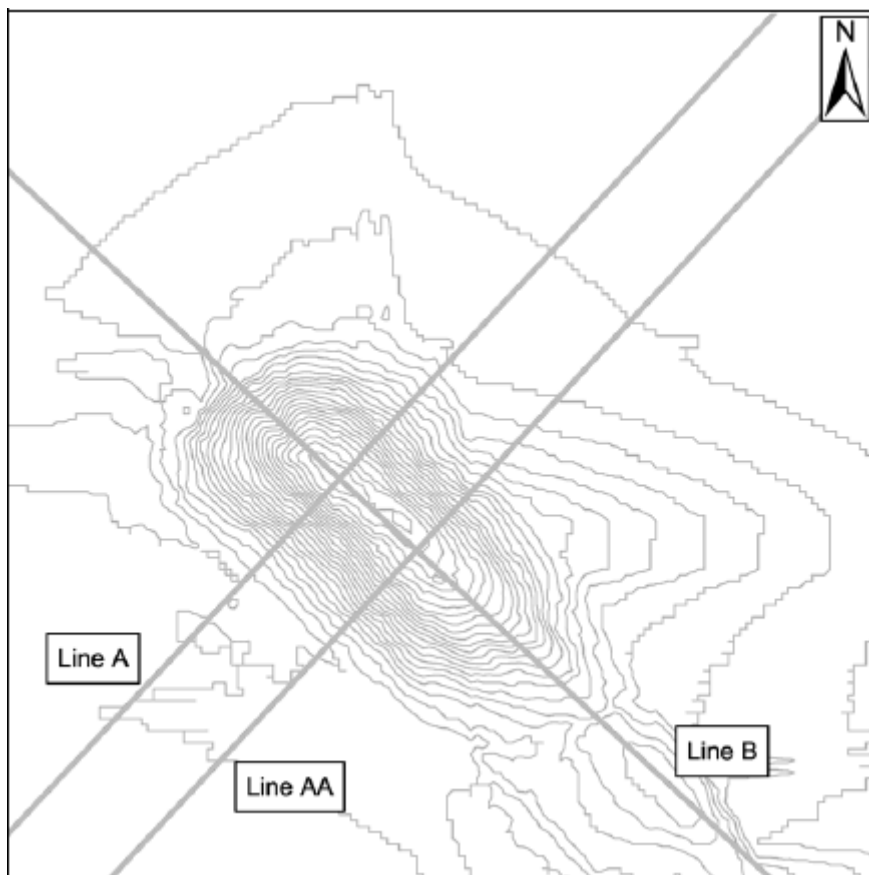


Fig 5.1- Mapa en el que se enseña la posición de las líneas [20]

En las figuras que se encuentran a continuación se observan las velocidades medidas y las obtenidas, los círculos son las velocidades medidas reales, las 'x' las velocidades

medidas con un mallado fino y el signo ‘+’ representa las velocidades medidas con un mallado más gordo. Como se puede observar no existe una diferencia considerable entre realizar el modelo con un mallado fino o hacerlo con uno gordo.

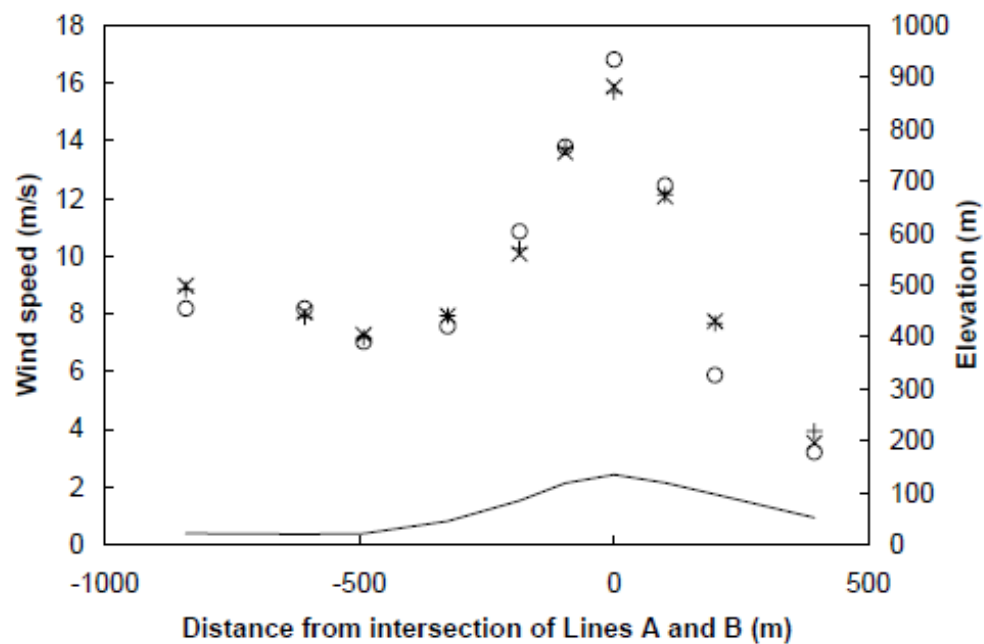


Fig 5.2- Velocidad a lo largo de la línea A (ver Fig 5.1) [20]

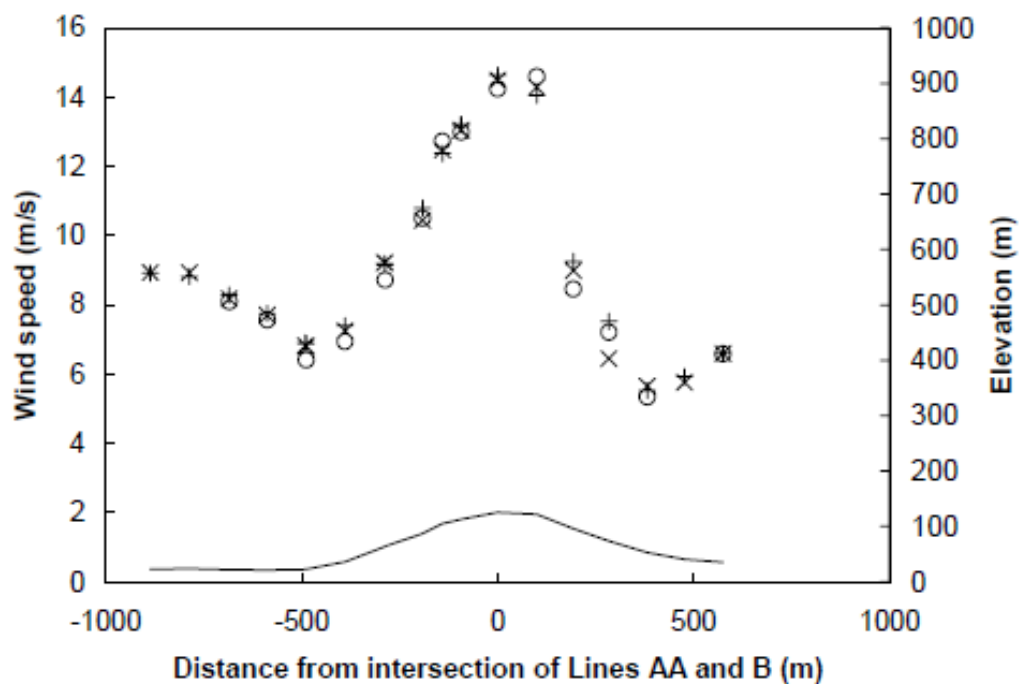


Fig 5.3- Velocidad a lo largo de la línea AA (ver Fig 5.1) [20]

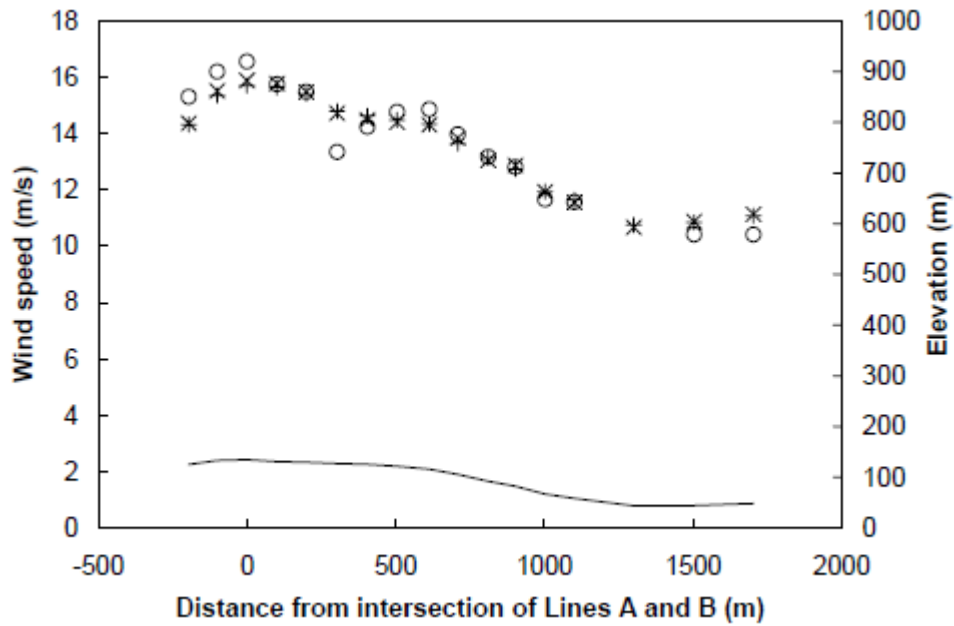


Fig 5.4- Velocidad a lo largo de la línea B (ver Fig 5.1) [20]

5.2 Comparación de los datos obtenidos por Forthofer con los del TFG

En este apartado se va a realizar una comparativa entre los datos obtenidos por Forthofer, los datos reales y los obtenidos usando la aplicación.

Para realizar esta comparativa se ha seguido el esquema representado en el diagrama de flujo inferior:

Diagrama de flujo del proceso de comparación

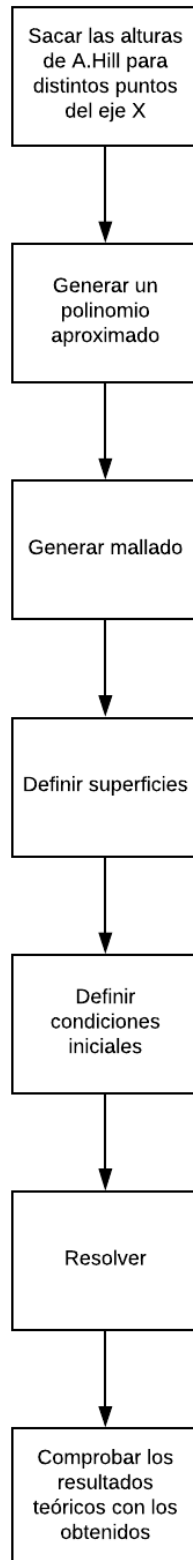


Fig 5.5- Diagrama de flujo de la comprobación de resultados

A continuación se explica brevemente el proceso seguido para la obtención de la comparación

5.2.1 Generar un polinomio aproximado

A partir de las alturas en las imágenes se sacan las distintas funciones de la montaña. Es importante destacar la dificultad que presenta el no tener la montaña como dato computacional, sino tener que obtenerla a partir de una gráfica, lo que puede provocar que las soluciones sean menos exactas

Para que la función se adapte lo mejor posible a la montaña el punto de mayor altura se debe poner ligeramente más alto, de esta manera se consigue que no se suavice más de lo necesario la forma de la montaña.

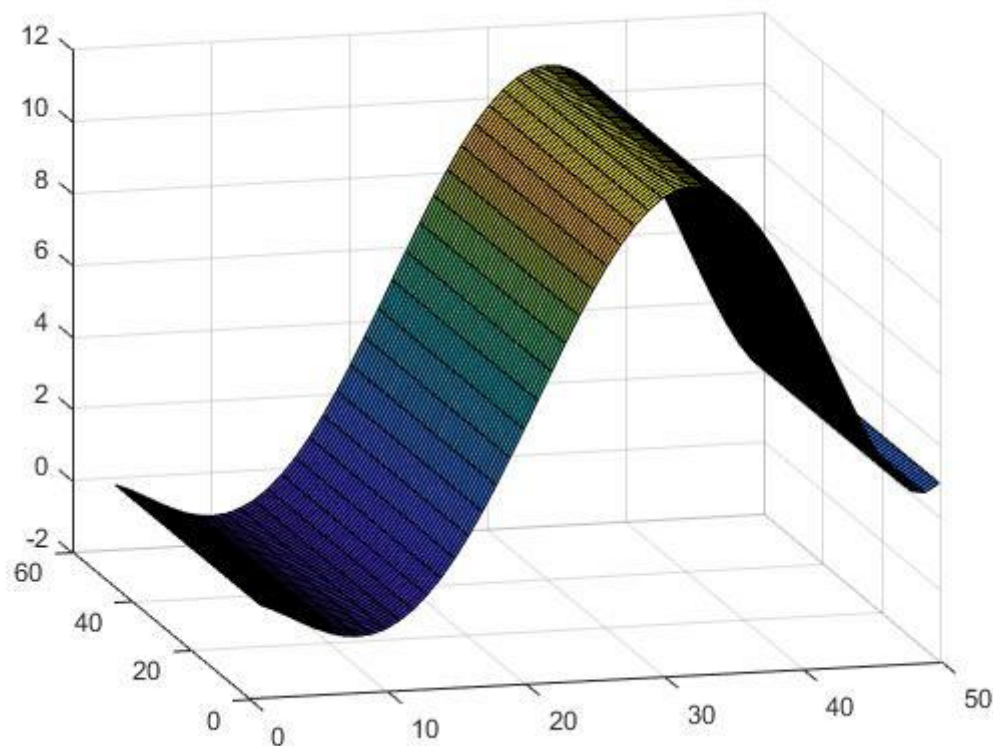


Fig 5.6- Ejemplo de superficie definida

5.2.2 Generar el mallado y definir las superficies

Se genera el mallado, y se definen las distintas superficies (entrada, pared lateral y salida). Para que sea computacionalmente posible realizar el mallado, es necesario escalar la función (se debe hacer 10 veces más pequeña). Es fundamental realizar un mallado adecuado para que la simulación sea lo más correcta posible

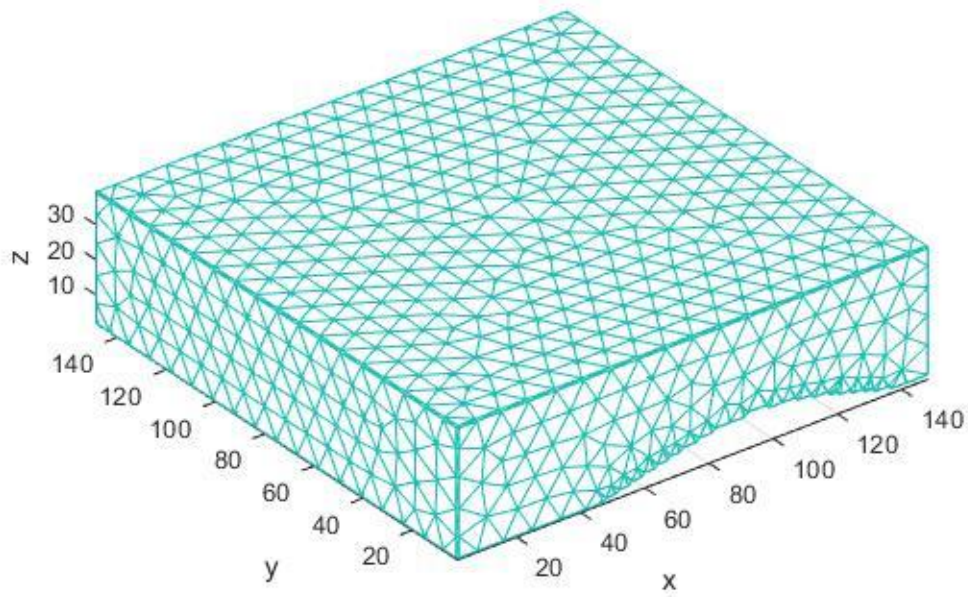


Fig 5.7- Ejemplo del mallado

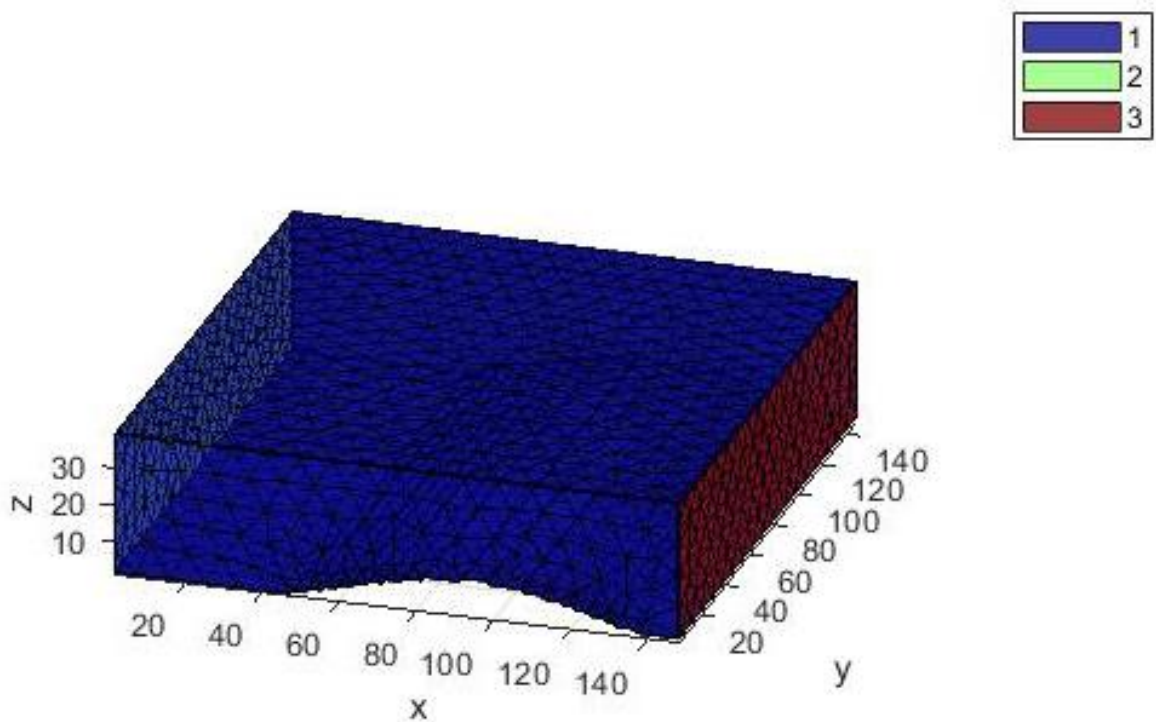


Fig 5.8- Mallado con superficies ya definidas

5.2.3 Resolución y comparativa de los resultados

Una vez se han definido las superficies se introduce la velocidad inicial (en estos casos 8.9 m/s), se resuelve el problema y se muestran los resultados. En la aplicación es necesario medir las velocidades a alturas ligeramente por encima de los 10m (entre 20 y 30), debido a la escala realizada. No se puede hacer el mallado lo suficientemente fino como para que el cambio de velocidad se produzca a los 10m.

5.3 Comparativas:

5.3.1 Línea A (Ver Fig 5.1 y Fig 5.2)

Los puntos utilizados para representar la función son los siguientes (sacados a partir de las gráficas de Forthofer):

X	-800	-600	-500	-300	-150	-100	0	100	180	400
Y	10	10	10	50	90	109	117	110	100	50

Tabla 5.1- Puntos utilizados para sacar la función

La función que más se adapta a estos puntos es un polinomio de grado 7 con los siguientes coeficientes:

- $p1 = -9.3649e-12$
- $p2 = -1.3323e-09$
- $p3 = -8.8229e-09$
- $p4 = 4.7968e-06$
- $p5 = 4.3153e-05$
- $p6 = -0.0093411$
- $p7 = 0.024668$
- $p8 = 11.814$

Que es la siguiente:

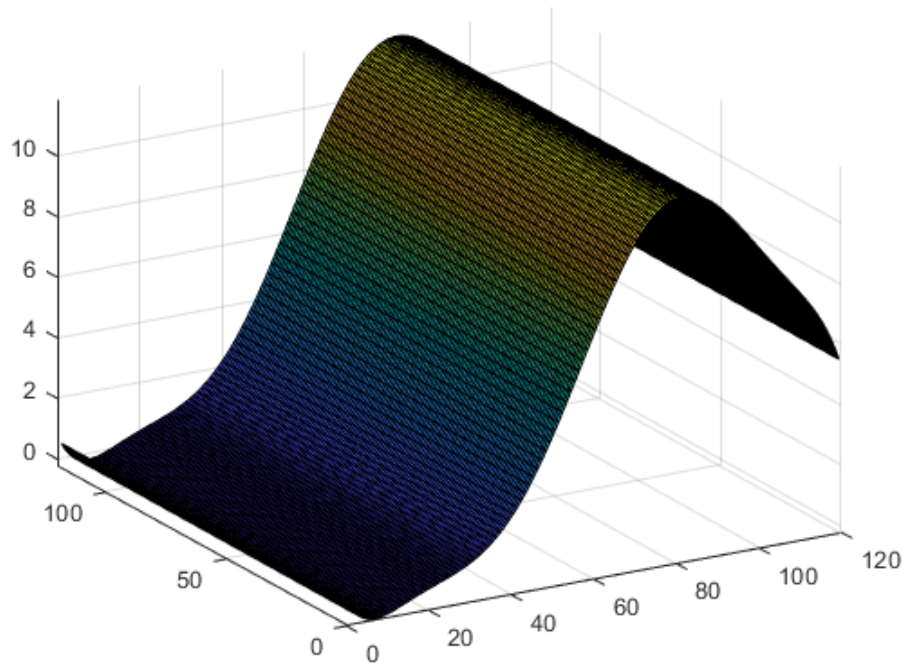


Fig 5.9- Superficie de la función de puntos A-A

Una vez definida la función se genera el mallado y se definen las superficies de este:

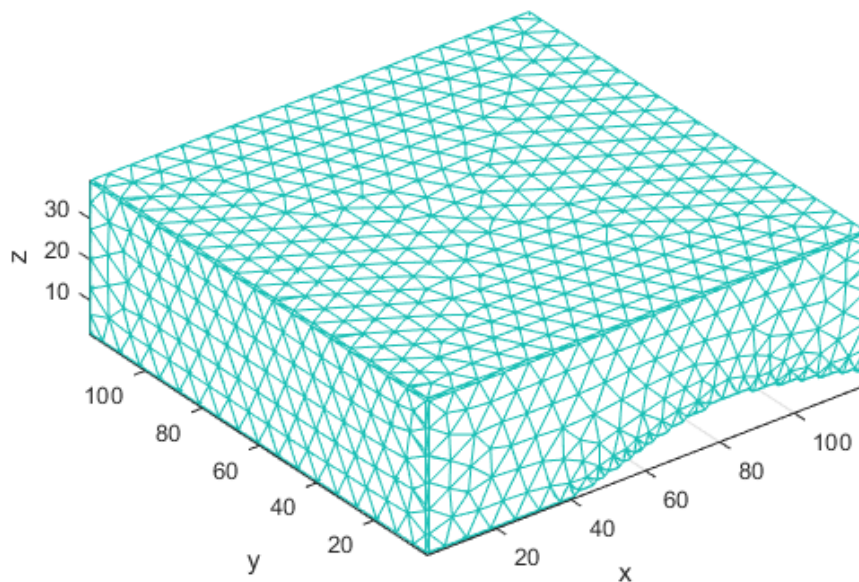


Fig 5.10- Mallado de A

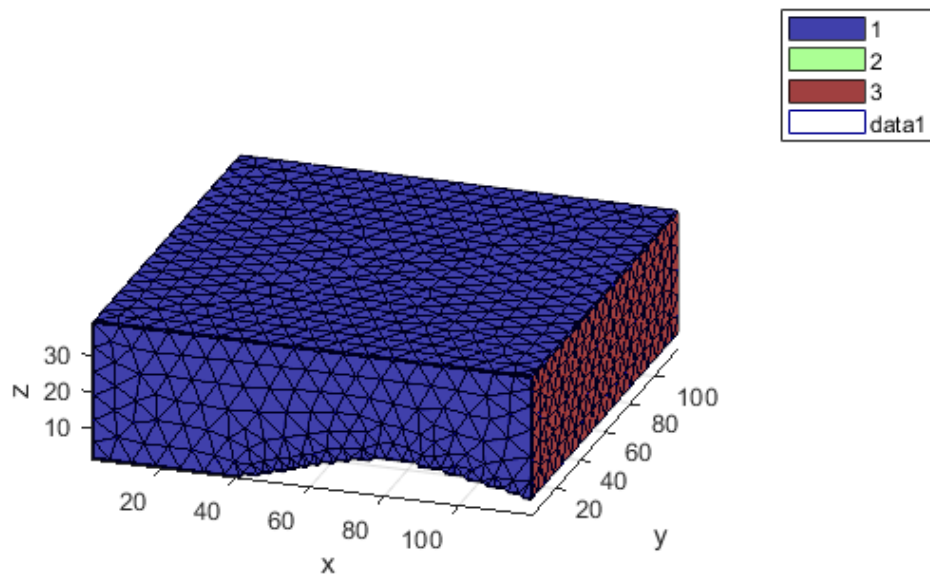


Fig 5.11- Mallado con superficies de A

Ya se puede resolver el problema, se muestra a continuación el perfil de velocidades obtenido

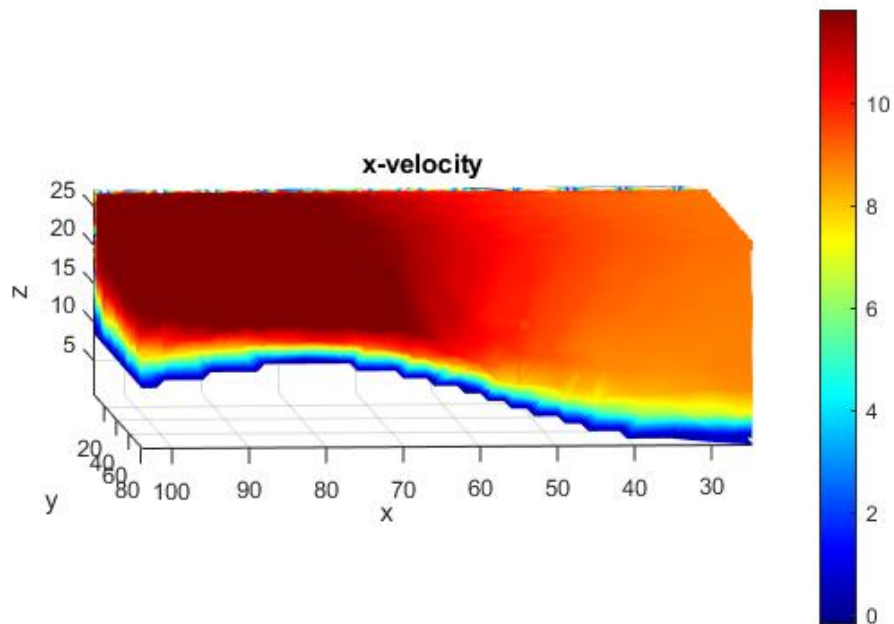


Fig 5.12- Perfil de velocidades de A

Aquí se puede ver la gráfica de la Fig 5.2 añadiendo la velocidad obtenida por nuestro programa. La mayor desviación se encuentra en los puntos más altos

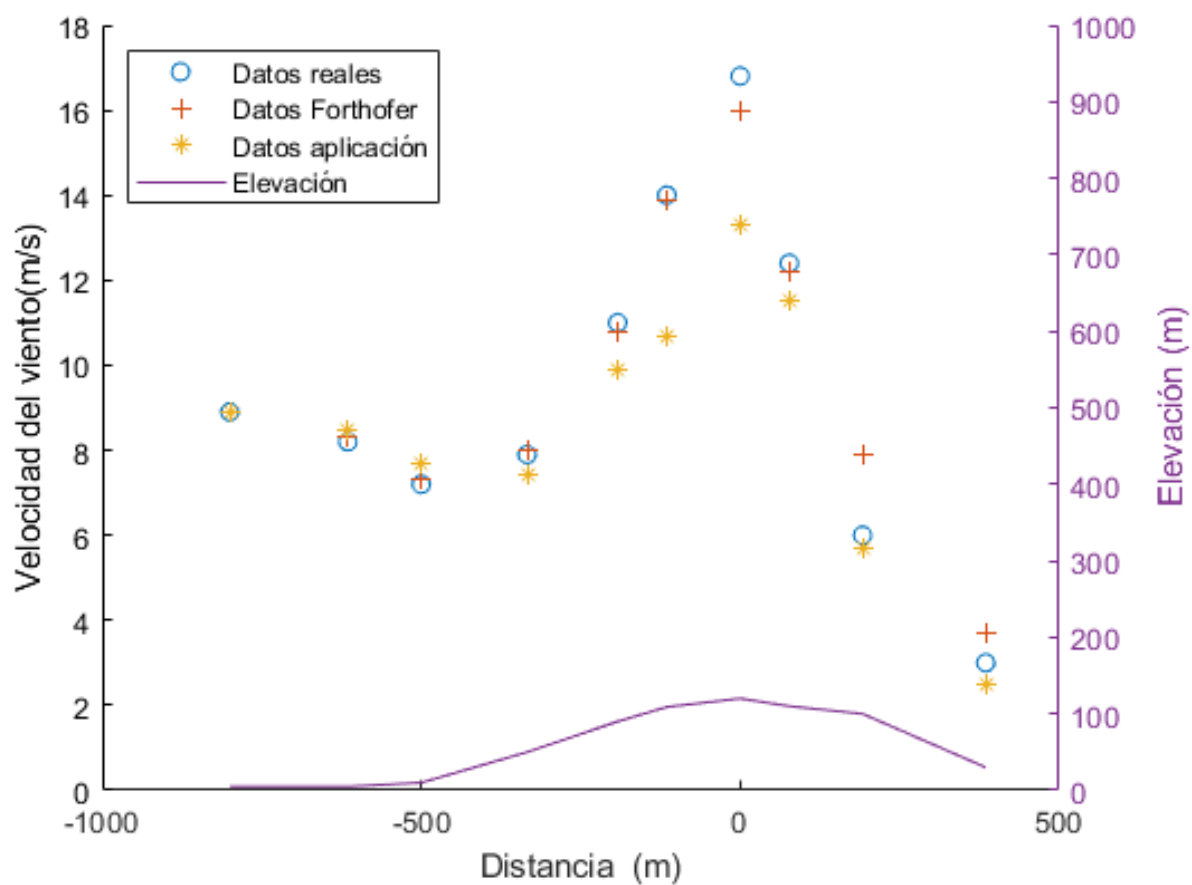


Fig 5. 13- Velocidad a lo largo de la línea A, ver Fig 5.1

Velocidad en los siguientes puntos (x) en m/s	Datos reales	Datos Forthofer	Datos simulación	Error(valor absoluto)	Error (porcentaje)
-800	8.9	8.9	8.9	0	0
-615	8.2	8.3	8.5	0,3	3,66
-500	7.2	7.3	7.7	0,5	6,94
-334	7.9	8	7.4	0,5	3
-192	11	10.8	9.9	1,1	10
-115	14	13.9	10.7	3,3	23,57
0	16.8	16	13.3	3,5	20,833
77	12.4	12.2	11.5	0,9	7,26
192	6	7.9	5.7	0,3	5
385	3	3.7	2.5	0,5	16,67

Tabla 5.2-Resumen de los resultados y errores

Como se puede observar, el error máximo se obtiene en la parte más alta de la montaña. Esto probablemente se deba a que la aplicación genera la superficie de Askervill Hill a partir de unos puntos, y aunque el polinomio se adapta bastante bien, en general tiende a ser más suave y menos exacta, lo que provoca una disminución de la velocidad máxima.

Por otra parte, es interesante destacar que las simulaciones realizadas con el código del TFG obtienen mejores predicciones que las de Forthofer en la ladera de la montaña en la que no incide el viento de manera directa.

5.3.2 Línea AA (Ver Fig 5.1 y Fig 5.3)

Los puntos utilizados para representar la función son los siguientes (sacados a partir de las gráficas de Forthofer):

X	-900	-600	-500	-300	-150	-100	0	50	100	200	400	580
Y	5	5	5	55	100	109	120	112	105	100	50	30

Tabla 5.3- Puntos utilizados para sacar la función

La función que más se adapta a estos puntos es un polinomio de grado 5 con los siguientes coeficientes:

- $p1 = 7.3816e-14$
- $p2 = 9.783e-11$
- $p3 = -3.2212e-08$
- $p4 = -6.2288e-05$
- $p5 = 0.0050588$
- $p6 = 11.469$

$$z = p1*X.^5 + p2*X.^4 + p3*X.^3 + p4*X.^2 + p5*X + p6$$

Que es la siguiente:

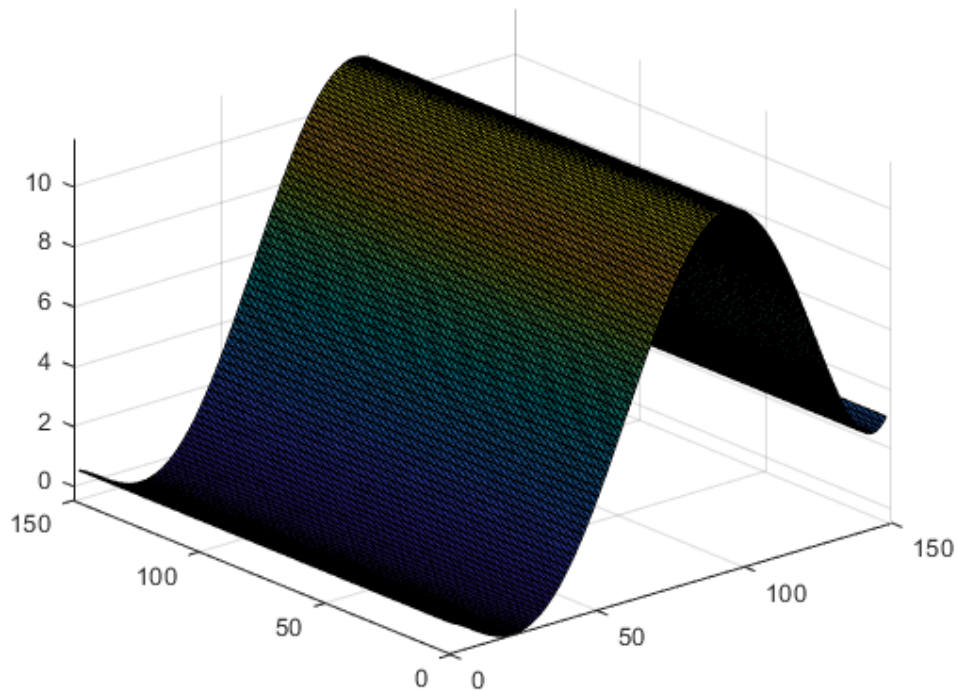


Fig 5.14- Función de puntos AA

Una vez definida la función se genera el mallado y se definen las superficies de este:

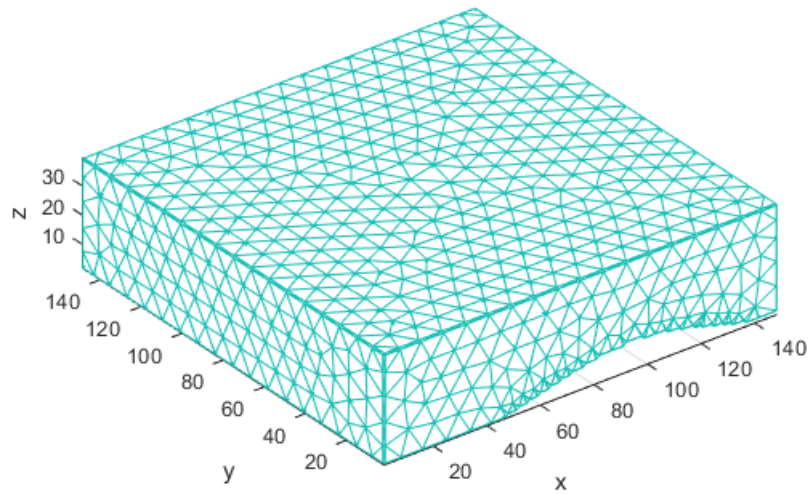


Fig 5.15- Mallado de AA

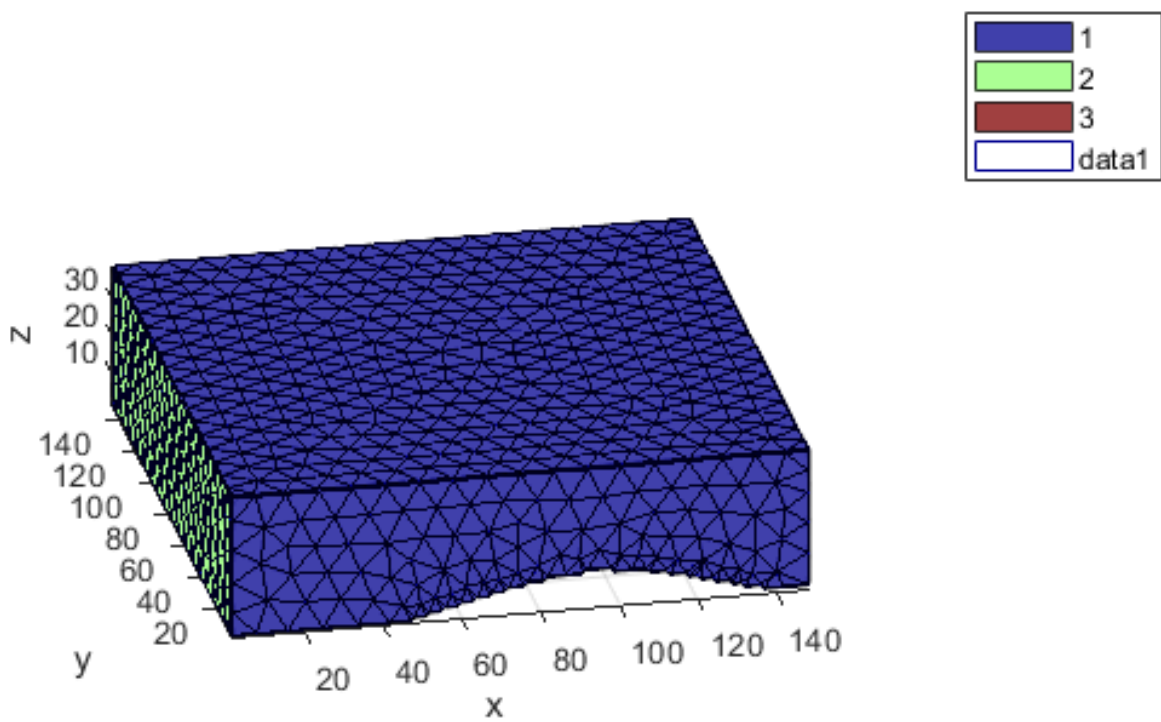


Fig 5.16- Mallado con superficies de AA

Ya se puede resolver el problema. Se muestra a continuación el perfil de velocidades obtenido

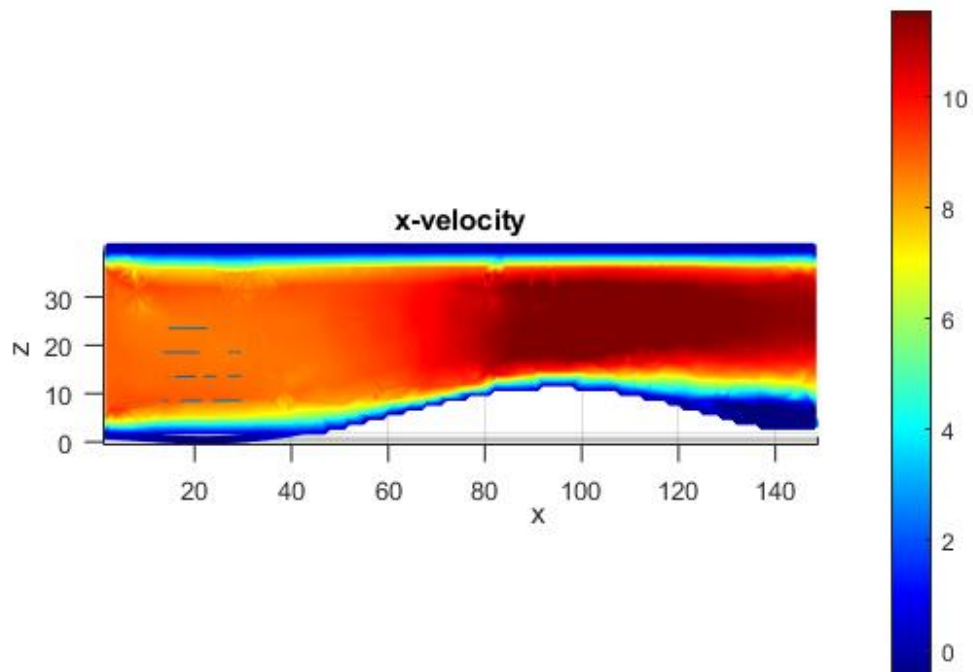


Fig 5.17- Perfil de velocidades de AA

Aquí se puede visualizar la gráfica de la Fig 5.3 añadiendo la velocidad obtenida por nuestro programa.

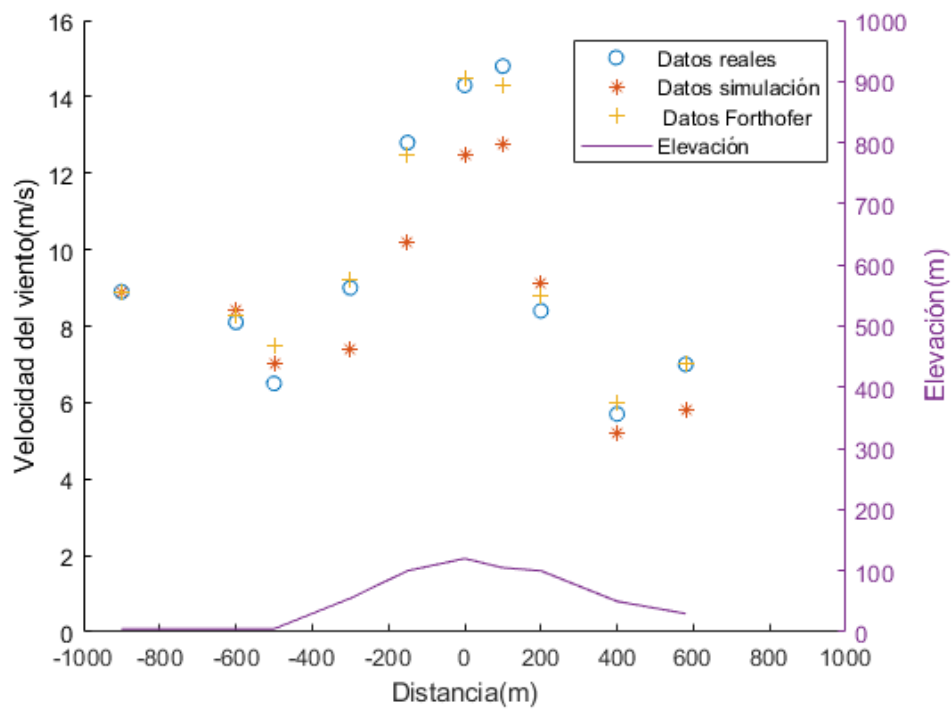


Fig 5.18- Velocidad a lo largo de la línea AA (ver Fig 5.3)

Velocidad en los siguientes puntos (x) en m/s	Datos reales	Datos Forthofer	Datos simulación	Error (valor absoluto)	Error (porcentaje)
-900	8.9	8.9	8.9	0	0
-680	8.1	8.3	8.4	0,3	3,70
-500	6.5	7.5	7	0,5	7,69
-300	9	9.2	7.4	1,6	17,78
-180	10.8	10.8	9	1,8	16,67
-150	12.8	12.5	10.8	2	15,6
-100	13	13.2	11	2	15,34
0	14.3	14.5	12.5	1,8	12,58
100	14.8	14.3	12.78	2,02	13,65
180	8.4	8.8	9.1	0,7	8,33
400	5.7	6	5.2	0,5	8,78
580	7	7	5.8	1,2	17.14

Tabla 5.4- Resumen de los resultados y errores

Si se realiza la simulación en esta ladera, los errores en valor absoluto son bastante menores. Esto se debe a que el polinomio se adapta mejor que el anterior. En esta simulación el error máximo en valor absoluto es de 2 m/s, y porcentualmente de 17.8%. Es especialmente interesante resaltar los buenos resultados obtenidos en el punto más alto de la montaña en comparación con los del ejemplo anterior.

5.3.3 Línea B (Ver Fig 5.1 y Fig 5.4)

Los puntos utilizados para representar la función son los siguientes (sacados a partir de las gráficas de Forthofer):

X	-200	0	300	500	700	1000	1275	1500	1700
Y	105	120	116	116	100	50	15	20	25

Tabla 5.5- Puntos utilizados para sacar la función

La función que más se adapta a estos puntos es un polinomio de grado 7 con los siguientes coeficientes:

- $p1 = 1.0002e-20$
- $p2 = -8.1084e-17$
- $p3 = 2.246e-13$
- $p4 = -2.6087e-10$
- $p5 = 1.1585e-07$
- $p6 = -1.3239e-05$
- $p7 = -0.0022823$
- $p8 = 11.995$
-

$$z = p1 * X.^7 + p2 * X.^6 + p3 * X.^5 + p4 * X.^4 + p5 * X.^3 + p6 * X.^2 + p7 * X + p8$$

Que es la siguiente:

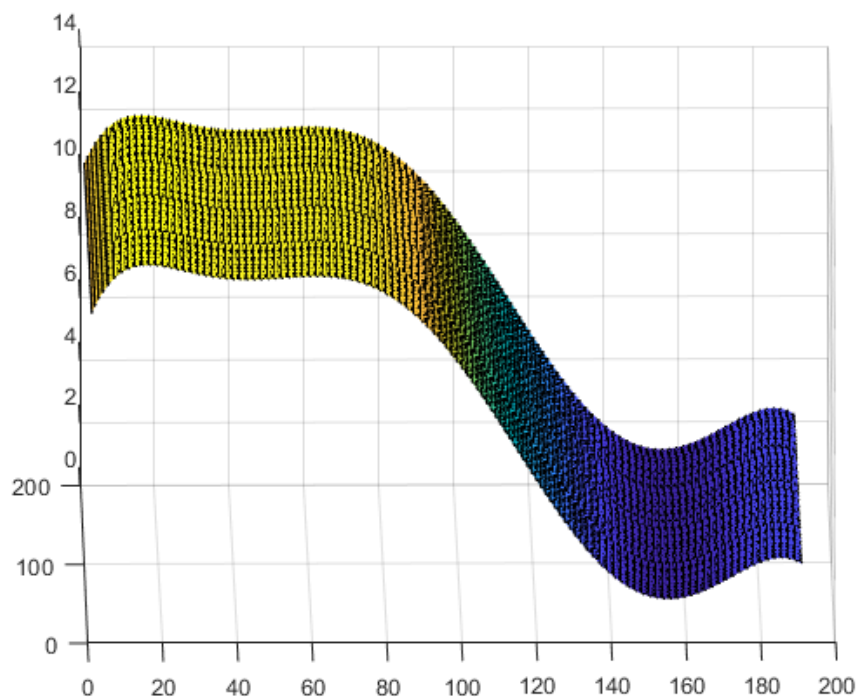


Fig 5.19- Función de B

Una vez definida la función se genera el mallado y se definen las superficies de este:

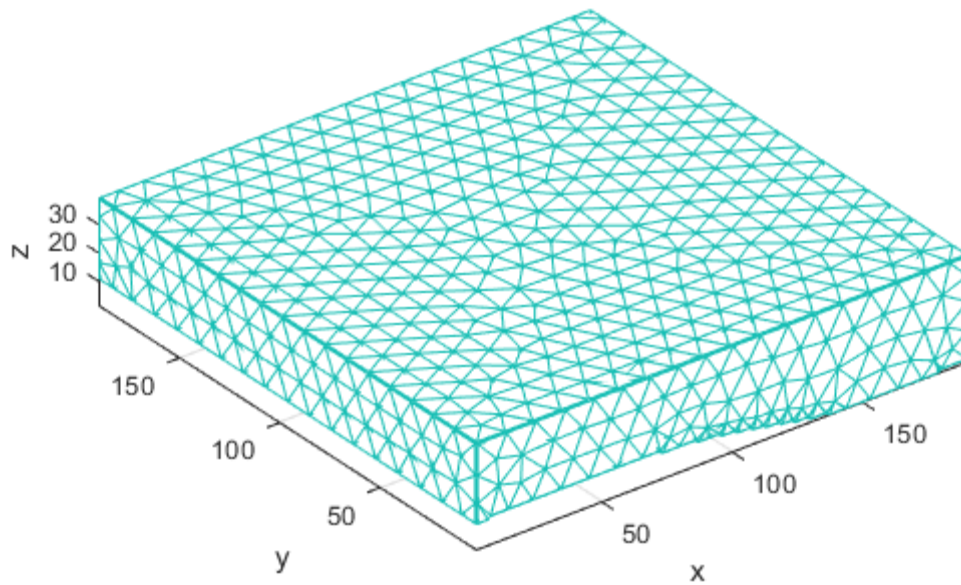


Fig 5.20- Mallado de B

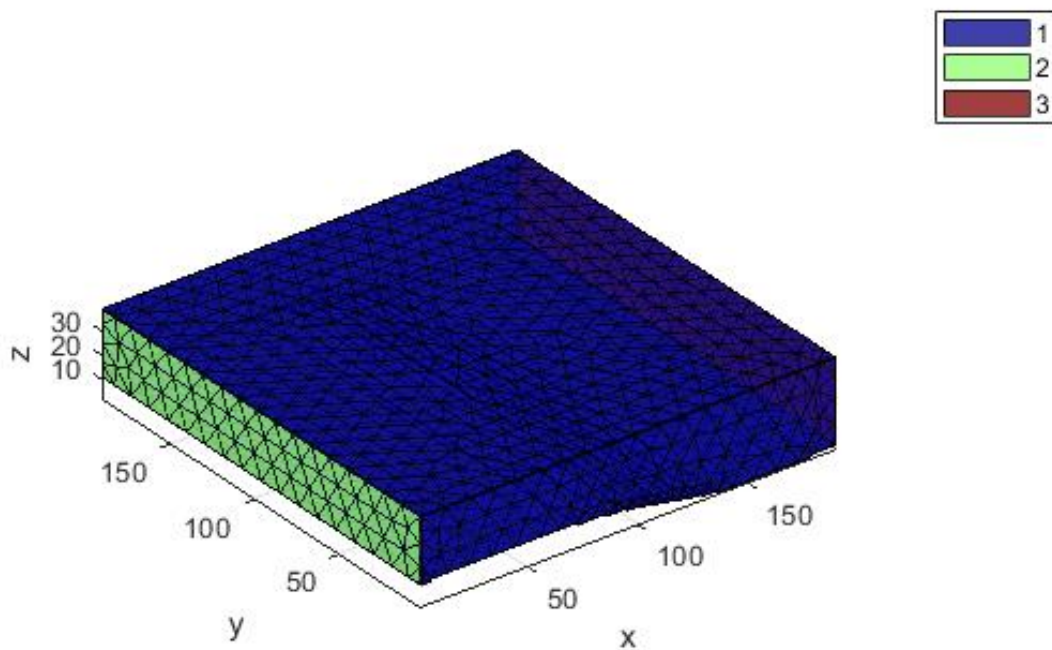


Fig 5.21- Mallado con superficies de B

Ya se puede resolver el problema. Se muestra a continuación el perfil de velocidades obtenido

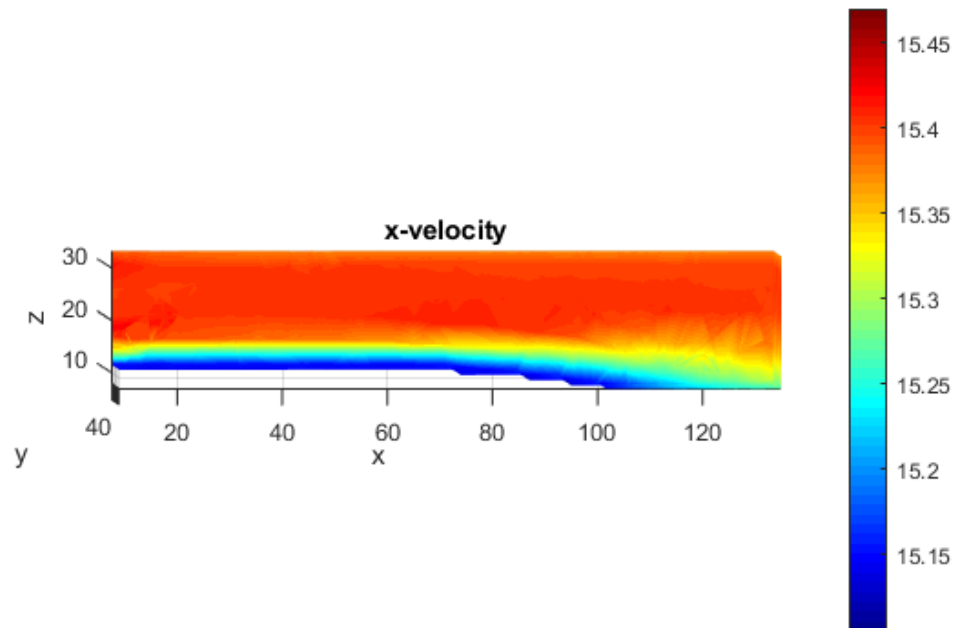


Fig 5.22- Perfil de velocidades de C

Aquí se puede visualizar la gráfica de la Fig 5.4 añadiendo la velocidad obtenida por nuestro programa.

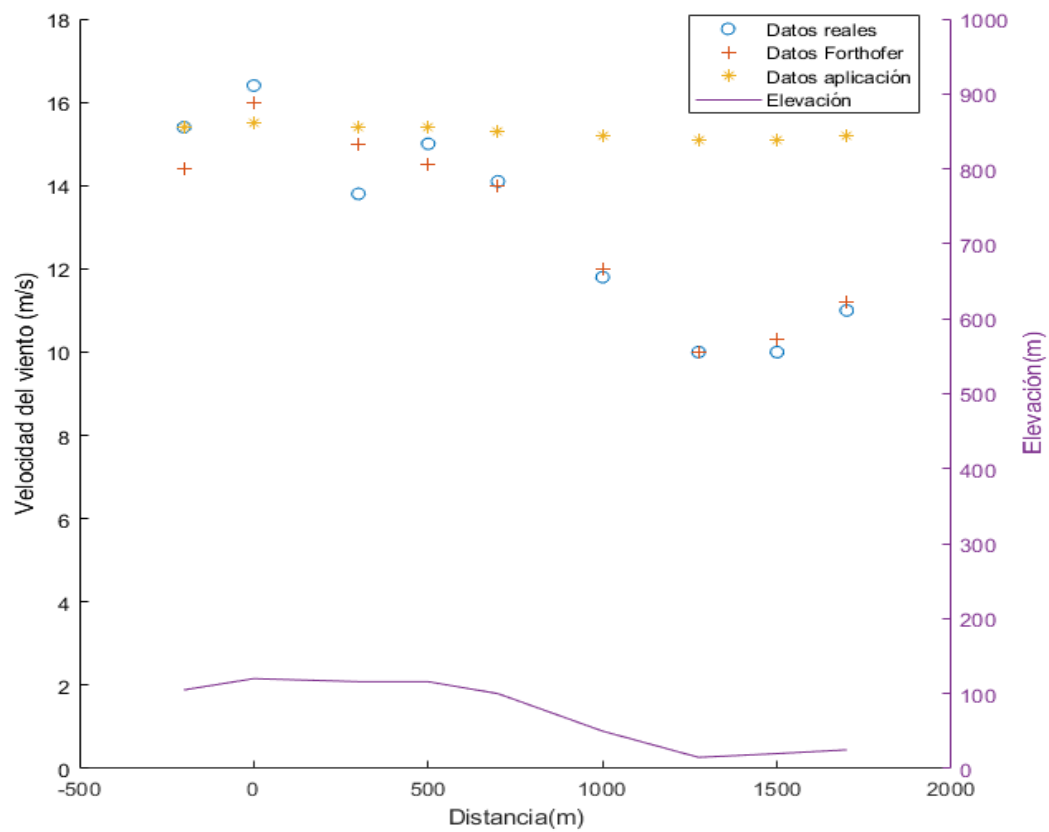


Fig 5. 23- Velocidad a lo largo de la línea B (ver Fig 5.4)

Velocidad en los siguientes puntos (x) en m/s	Datos reales	Datos Forthofer	Datos simulación	Error(valor absoluto)	Error(porcentaje)
-200	15.4	14.4	15.4	0,9	5,487
0	16.4	16	15.5	1,6	11,594
300	13.8	15	15.4	0,4	2,667
500	15	14.5	15.4	1,2	8,511
700	14.1	14	15.3	3,4	28,813
1000	11.8	12	15.2	5,1	51
1275	10	10	15.1	5,1	51
1500	10	10.3	15.1	4,2	38,182
1700	11	11.2	15.2	0,9	5,488

Tabla 5.6- Resumen de los resultados y errores

En este caso, se obtienen, sin lugar a dudas los peores resultados. Esto se debe a que el terreno apenas presenta pendiente. Cuando se obtiene la función a partir de los puntos, esta tiende a aplanarse aún más que el terreno original. Esto provoca que el programa resuelva un terreno prácticamente plano, con lo que la velocidad apenas decrece. Para este tipo de zonas sería conveniente realizar las simulaciones en condiciones de flujo laminar, pero computacionalmente no es posible.

6. Conclusiones: análisis y mejoras

6.1 Conclusión

El objetivo de este Trabajo de Fin de Grado es desarrollar una aplicación en Matlab que mejore los sistemas de predicción de incendios existentes. También se desea continuar la línea de investigación abierta en la Universidad Carlos III de Madrid sobre la predicción del desarrollo de incendios forestales.

Una vez realizado el estudio se busca comprobar si los resultados son lo suficientemente buenos o no, para continuar con el tratamiento del problema con el uso de técnicas CFD o descartar este enfoque en futuros trabajos.

Una vez analizado, se debe resaltar, que este método obtiene muy buenos resultados. No mejoran a los obtenidos por Forthofer en cuanto a precisión, pero sí lo hacen en cuanto a la capacidad para resolver distintas geometrías. Por todo ello este programa está especialmente indicado para dar soluciones aproximadas de manera rápida en distintos terrenos geográficos.

De este modo, a continuación se realiza un análisis crítico de los resultados, del entorno socioeconómico y del marco regulador. Del mismo modo se exponen mejoras y trabajo futuro, de cara a posteriores investigaciones.

6.2 Análisis crítico de los resultados

Tras el desarrollo y prueba de la aplicación, es necesario hacer un análisis de las ventajas e inconvenientes de esta.

- **Ventajas:** Uno de los puntos fuertes de la aplicación es el ser capaz de hacer frente a distintos problemas con cambios mínimos en el código, lo que es fundamental. De esta manera cualquier persona puede hacer uso de ella simplemente leyendo unas instrucciones. En segundo lugar, es especialmente destacable el hecho de que se pueda obtener la velocidad prácticamente en cada punto, con una gran precisión. Esto es fundamental para saber exactamente en qué zonas el incendio va a presentar mayor agresividad y peligro.
- **Inconvenientes:** El principal inconveniente de esta aplicación reside en la dificultad que presenta el representar los vectores de velocidad en geometrías complejas, quedando en ocasiones demasiado poco visuales pese al tratamiento dado. Por otra parte, en geometrías complejas el tiempo de cálculo es muy alto, lo que supone indudablemente un gran problema, debido a que en caso de incendio, el tiempo es un factor fundamental.

6.3 Entorno socioeconómico

Este TFG podría tener impacto inmediato al colaborar en la prevención de incendios, reduciendo los tiempos de extinción. Esto generaría un gran impacto medioambiental, al reducirse el terreno quemado. Evidentemente un menor impacto medioambiental trae consigo un menor impacto económico, por las grandes pérdidas económicas que generan los incendios.

Presupuesto:

En la tabla que se muestra a continuación, se muestra el presupuesto que ha sido necesario para realizar el TFG. Al tratarse de un proyecto realizado únicamente a ordenador, el coste del proyecto viene determinado por el precio de las licencias de los programas y las horas de trabajo:

Descripción	Unidades	Precio unitario	Coste(€)
Licencia Académica individual Matlab	1	500	500
Licencia Académica individual QuickerSim	1	169	169
Horas trabajadas	450	10	4500
Total neto			5169
IVA(21%)			1085
Total			6254

Como se puede observar, el coste es bastante bajo. Si este se compara con la cantidad de dinero que invierte España en la extinción de incendios se comprueba que el proyecto supone un desembolso casi nulo. Si bien las mejoras a introducir pueden elevar ligeramente los costes, sigue siendo una vía a explorar muy rentable por los beneficios económicos, sociales y medioambientales que este programa puede producir.

6.4 Marco regulador

En el siguiente marco regulador, se hace un pequeño análisis de la legislación aplicable sobre el uso de programas como Matlab para la prevención de incendios forestales, y sobre el propio programa trabajado y su aplicación.

Se trata de un proyecto de desarrollo de una Toolbox capaz de predecir la propagación de un incendio, por lo que el código no tiene ninguna regulación legal específica, tan solo los derechos de propiedad intelectual de Matlab y QuickerSim.

El programa debe respetar la legislación actual en materia de prevención de incendios, es decir debe tener en cuenta cualquier legislación de carácter legal o económica como la “Ley 3/2010, de 18 de febrero, de prevención y seguridad en materia de incendios en establecimientos, actividades, infraestructuras y edificios”.

Por otro lado, queda comprobada la efectividad de este programa para la prevención de incendios forestales, tal y como se indica anteriormente y se puede revisar en las comparativas.

6.5 Mejoras y trabajo futuro

Los siguientes aspectos son aquellos que podrían modernizar el programa, mejorando sus prestaciones.

Velocidades de viento bajas: Como se ha mencionado anteriormente, los programas CFD no son capaces de trabajar con velocidades muy bajas con exactitud. Esto se debe, a que este código está diseñado para flujo turbulento, lo que provoca que sea exacto a velocidades ligeramente altas. Es posible definir el problema como flujo laminar. Sin embargo, esto limita mucho las geometrías, tardando tiempos excesivamente altos en resolver geometrías sencillas y siendo incapaz de resolver geometrías complejas.

Aumento de la velocidad de cálculo: En caso de incendio la velocidad es un factor fundamental. A mayor velocidad en la obtención de soluciones, se puede intervenir y preparar la situación antes, lo que suele ser decisivo. En nuestro caso, el ordenador usado para realizar las simulaciones tiene un procesador Intel I3. En geometrías complejas cada simulación tarda varias horas en flujo turbulento y en simulaciones realizadas con velocidades bajas (flujo laminar) cada simulación puede llegar a tardar varios días. Esto, es evidentemente inadmisibile a la hora de predecir el desarrollo de un incendio.

Inclusión de nuevos parámetros: En este Trabajo de Fin de Grado solo se está tratando la trayectoria del viento en función de la geografía. La extensión del incendio depende de muchos más parámetros, y para evaluar la extensión real del fuego habría que tener en cuenta todos ellos. Sería interesante realizar otro TFG que desarrollase el código incluyendo el resto de variables. De esta forma se obtendría uno de los programas más completos de predicción.

Optimización del código: En ocasiones la aplicación es realmente lenta, esto probablemente, pueda solucionarse además de utilizando un ordenador con mayor capacidad de cálculo mediante la optimización del código, especialmente en las partes de representación gráfica.

Estética del código: Es posible introducir cambios en el código que provoquen que sea más agradable a la vista y más sencillo de leer.

Automatización de los inputs: Esto realmente daría para un TFG en sí mismo. Sería interesante que introduciendo un mapa, el programa redimensionase la matriz lo necesario de manera automática, sin necesidad de que el usuario intervenga. Así como que el programa sea capaz de establecer los límites de altura de la malla.

Por otra parte a la hora de visualizar los resultados, el usuario tiene que elegir los puntos en los que comienza la trayectoria o los puntos en los que desea ver los vectores. Si el programa realizase esto en las zonas superficiales de manera autónoma, supondría una gran mejora, dado el tiempo que requiere para el usuario realizar esto.

También sería interesante que el programa obtuviese de manera automática y con mayor precisión la velocidad de los puntos pertenecientes a una trayectoria, sin necesidad de que el usuario busque la localización de cada punto de manera particular.

Personalización del código de QuickerSim: Los desarrolladores de la Toolbox dan la posibilidad de desarrollar un software específico para un problema determinado. Evidentemente un software específico proporcionaría mejores resultados en la resolución de las ecuaciones de Navier-Stokes, generando unas soluciones más cercanas a las reales. En definitiva, sería necesario implementar una programación específica en caso de que se tuviera intención de comercializar el programa, lo que también supondría costes adicionales.

Bibliografía y Webgrafía

1. L. Hernández, “El polvorín del noroeste. Propuesta ibérica de WWF España y ANP/ WWF Portugal para la prevención de incendios”, WWF, España, **Informe Técnico**, 2018. [En línea] Disponible en: http://awsassets.wwf.es/downloads/informe_incendios_forestales_wwf_2018.pdf?_ga=2.19627128.1393140619.1552671281-1859981087.1552671281
2. Ministerio para la Transición Ecológica. Página web del ministerio. URL: <https://www.miteco.gob.es/es/> [acceso: 3 de mayo de 2019]
3. R. Garrido, R., Carreira y R. Iglesias, *Defensa y prevención de incendios forestales*. Madrid: Síntesis S.A, 2016. [En línea] Disponible en: <https://www.sintesis.com/data/indices/9788490773062.pdf>
4. I. Fernández Migueláñez, “Evaluación de la eficacia para la prevención de incendios del aprovechamiento energético de la biomasa forestal en el macizo de Caroig (Valencia, España)”, **Trabajo Fin de Máster**, Facultad de Geografía e Historia, Universidad Complutense de Madrid, España, 2013. [En línea] Disponible en: https://eprints.ucm.es/23804/2/TFM_Irma_Fernandez_Miguela%C3%B1ez.pdf
5. D. Alvear, G. Rein, M. Lázaro y J.A. Capote, *Modelado y simulación computacional de incendios en la edificación*. Santander: Ediciones Díaz Santos, 2007. [En línea] Disponible en: <http://www.editdiazdesantos.com/wwwdat/pdf/9788479788322.pdf>
6. C. Nicolás Martín, “Modelización de un Incendio Forestal con Fast Marching”. **Trabajo Fin de Grado**. Ingeniería en Tecnologías Industriales, Universidad Carlos III de Madrid, España, 2007. [En línea] Disponible en: <https://e-archivo.uc3m.es/handle/10016/27616>
7. Centro de investigación forestal Lourizán. **Página web**. URL: <https://lourizan.xunta.gal/> [acceso: 3 de mayo de 2019]
8. J. A. Capote, “Limitaciones de la simulación computacional en la Protección contra incendios en la edificación”. **Trabajo de Investigación**. GIDAI, Universidad de Cantabria, España, 2018. [En línea] Disponible en: <https://grupos.unican.es/gidai/web/investigacion/Trabajo%20Investigacion%20MSCI.pdf>
9. A. Bargueño Juárez, “Modelización de un Incendio Forestal mediante Fast Marching”. **Trabajo Fin de Máster**. Ingeniería Industrial, Universidad Carlos III de Madrid, España, 2016.
10. R. González Pacheco, “Perfeccionamiento de un simulador de incendios forestales basado en Fast Marching”. **Trabajo Fin de Grado**. Ingeniería en Tecnologías Industriales, Universidad Carlos III de Madrid, España, 2018.

11. Programa “Interreg Sudoe”. **Página web**. URL: <http://www.fire-rs.com/es/> [acceso: 3 de mayo de 2019]
12. C. Tymstra, R.M. Bryce, B.M. Wotton, S.W. Taylor y O.B. Armitage, “Development and Structure of Prometheus: the Canadian Wildland Fire Growth Simulation Model” **Information Report NOR-X-417**. Northern Forestry Centre, 2010. [En línea] **Disponible en:** http://www.cfs.nrcan.gc.ca/bookstore_pdfs/31775.pdf
13. Wikipedia, “Principio de Fresnel- Huygens”. En la página web de Wikipedia. URL del recurso: https://es.wikipedia.org/wiki/Principio_de_Fresnel_-_Huygens [acceso: 3 de mayo de 2019]
14. B. W. Butler, M. Finney, L. Bradshaw, J. Forthofer, C. McHugh, R. Stratton y D. Jimenez, “WindWizard: A New Tool for Fire Management Decision Support”. *USDA Forest Service Proceedings RMRS-P-41*. 2006. [En línea] **Disponible en:** https://www.fs.fed.us/rm/pubs/rmrs_p041/rmrs_p041_787_796.pdf
15. M. Finney. “Fire, Fuel, Smoke Science Program. Rocky Mountain Research Station”. En la página web Firelab. URL del recurso: <https://www.firelab.org/project/farsite> [acceso 3 de mayo de 2019]
16. “Un nuevo software mejora el índice de prevención de riesgo de incendios” **Noticia de la Agencia EFE**. URL del recurso: <https://www.efe.com/efe/comunitat-valenciana/reportajes/un-nuevo-software-mejora-el-indice-de-prevencion-riesgo-incendios/50000901-3299011> [acceso: 4 de mayo de 2019]
17. Wikipedia, “Ecuaciones de Navier Stokes”. En la página web de Wikipedia. URL del recurso: https://es.wikipedia.org/wiki/Ecuaciones_de_Navier-Stokes [acceso: 3 de mayo de 2019]
18. X. Mora, “Las ecuaciones de Navier-Stokes ¿Impredictibilidad incluso sin mariposas?” **Página web Méthode**. URL del recurso: <https://methode.es/revistas-methode/monograficos/equacions-navier-stokes.html> [acceso 3 de mayo de 2019]
19. QuickerSim CFD Toolbox, **Página web**. URL: <https://quickersim.com/> [acceso: 4 de mayo de 2019]
20. J. M. Forthofer, “Modeling wind in complex terrain for use in fire spread prediction”, **Tesis Doctoral**. Department of Forest, Rangeland, and Watershed Stewardship, Colorado State University, EEUU, 2007. [En línea] **Disponible en:** https://www.firelab.org/sites/default/files/images/downloads/Forthofer_2007_thesis.pdf
21. LANDFIRE Data Distribution Site, **Página web**. URL: <https://landfire.cr.usgs.gov/viewer/> [acceso: 3 de mayo de 2019]

22. A. López Santalla, V. Jemes Díaz, y C. Roder Merino, “Evolución histórica y reciente de los grandes incendios forestales en España. Perspectivas de futuro”. **Presentada en** el 7º Congreso Forestal Español: Gestión del monte: servicios ambientales y bioeconomía. Plasencia, junio de 2017. **[En línea] Disponible en:** <https://7cfe.congresoforestal.es/sites/default/files/comunicaciones/426.pdf>
23. MathWorks, **Página web.** URL: <https://es.mathworks.com/> [acceso: 3 de mayo de 2019]

Anexos: Código

A.1 Código de Peaks

```
%% load the sample data
%% load the sample data
clear

Zaux=0.5*abs(peaks)+5;
[a,b]=size(Zaux);
Z2 = ones(a+8,b+8)*5;
for i=1:a
    for j=1:b
        Z2(i+4,j+4) = Zaux(i,j);
    end
end
[a,b]=size(Z2);

figure;
surf(Z2)
Z2=fliplr(Z2);
h=50;
W=zeros(a,b,h);
for i=1:a
    for j=1:b
        for k=1:h
            if k>=round(Z2(i,j))
                W(i,j,k)=1;
            end
        end
    end
end
W(:,1,:)=0;
W(1,:,:)=0;%izda
W(:,b,:)=0;
W(a,:,:)=0;%dcha

W(:,:h)=0;
W3=double(bwdist(W));

[X, Y, Z] = meshgrid(1:57,1:57,1:50);
[faces, vertices] = MarchingCubes(X, Y, Z, W3, .5);
model_filename = 'peaks_model.stl';
stlwrite(model_filename, faces, vertices);
model = createpde();
importGeometry(model, model_filename);
mesh = generateMesh(model, 'GeometricOrder', 'linear');
p = model.Mesh.Nodes;
t = model.Mesh.Elements;
t(end + 1, : ) = 1;
displayMesh(p, t);

triangulation_object = triangulation(t(1:end - 1, :)', p');
faces = triangulation_object.freeBoundary';
face_centers_X = (p(1, faces(1, :)) + p(1, faces(2, :)) + p(1,
faces(3, :)))/3;
x_min = min(face_centers_X);
x_max = max(face_centers_X);
```

```

inlet_faces = face_centers_X < x_min + 0.1;
outlet_faces = face_centers_X > x_max - 0.1;
e = [faces; ones(3, size(faces, 2))];
e(6, :) = 0;
e(4, inlet_faces) = 2;
e(4, outlet_faces) = 3;
viewCase

[p,e,t,nVnodes,nPnodes,indices] = convertMeshToSecondOrder(p,e,t);

displayMesh(p,t);

% Define fluid
nu = 1.5e-5;

% Init solution and convergence criteria

[u, convergence] = initSolution(p,t,[20 0 0],0);

maxres = 1e-3;
maxiter = 10;
alpha = 0.7;

% Compute wall distance and duplicate velocity vector
d = computeWallDistance(p,e,t,1);
us = u;

% Define inlet velocity profile
vel = [20 0 0];

% Iterate nonlinear terms
for iter = 1:maxiter
    % Compute turbulent viscosity
    nut = turbulentViscosity(p,t,u,d,nu,indices,'ChenXu');

    % Under-relax velocity and assemble matrix and right hand side
    u = alpha*u+(1-alpha)*us;
    [NS, F] =
assembleNavierStokesMatrix(p,e,t,nut,u(indices.indu),u(indices.indv
),u(indices.indw),'supgDoublyAsymptotic');

    % Apply boundary conditions
    [NS, F] = imposeCfdBoundaryCondition(p,e,t,NS,F,2,'inlet', vel);
    [NS, F] = imposeCfdBoundaryCondition(p,e,t,NS,F,1,'wall');

    % Compute and plot residuals
    [stop, convergence] =
computeResiduals(NS,F,u,size(p),convergence,maxres);
plotResiduals(convergence,2);

    % Break if solution converged
    if(stop)
        break;
    end

    % Solve equations
    us = u;
    %u = NS\F;

```

```

        [u,convergence] =
iterativeSolver(NS,F,u,indices,convergence,'Krylov');
end

vmag = sqrt(u(indices.indu).^2 + u(indices.indv).^2 +
u(indices.indw).^2);

% Plot solutio
figure(3);
displaySolution(p,t,u(indices.indu),'x-velocity');

vx=u(indices.indu)
vy=u(indices.indv)
vz=u(indices.indw)
velocidad=[vx;vy;vz]

points=p'
% Generate the grid of points where the interpolated values will be
calculated
[xg,yg,zg] = meshgrid(3.5 : 1 : 55);
% Create the interpolating object
F = scatteredInterpolant(points, vx);
% Do the interpolation
uu = F(xg,yg,zg);
% Now you can use isosurface with the gridded data
% Create the interpolating object
G = scatteredInterpolant(points, vy);
% Do the interpolation
v = G(xg,yg,zg);
% Now you can use isosurface with the gridded data
% Create the interpolating object
H = scatteredInterpolant(points, vz);
% Do the interpolation
w = H(xg,yg,zg);
% Now you can use isosurface with the gridded data
[sx,sy,sz] = meshgrid(5,5:10:45,6.3);
figure
    surf(Z2)
    demcmap(Z2,10000)
hold on
plot3(sx(:),sy(:),sz(:),'*r');
streamline(xg,yg,zg,uu,v,w,sx,sy,sz)
hold off

lims = [0 50 9 25 6 37];
[xsub,ysub,zsub,usub,vsub,wsb] = subvolume(xg,yg,zg,uu,v,w,lims);
figure(5)
    surf(Z2)
    demcmap(Z2,10000)
hold on
streamline(xsub,ysub,zsub,usub,vsub,wsb,sx,sy,sz)
hold off

figure(6)
surf(Z2)
demcmap(Z2,10000)
hold on

```



```

% Generate the grid of points where the interpolated values will be
calculated
[xg,yg,zg] = meshgrid(3.5 : 5: 45);
lims = [10 20 9 25 5 25];
figure(6)
    demcmap(Z2,10000)
surf(Z2)
hold on
% Create the interpolating object
F = scatteredInterpolant(points, vx);
% Do the interpolation
egx = F(xg,yg,zg);
% Now you can use isosurface with the gridded data

% Create the interpolating object
G = scatteredInterpolant(points, vy);
% Do the interpolation
egy = G(xg,yg,zg);
% Now you can use isosurface with the gridded data
% Create the interpolating object
H = scatteredInterpolant(points, vz);
% Do the interpolation
egz = H(xg,yg,zg);
[xsub,ysub,zsub,usub,vsub,wsub] =
subvolume(xg,yg,zg,egx,egy,egz,lims);

quiver3(xsub,ysub,zsub,usub,vsub,wsub,3)

```

A.2 Código para realizar pruebas en una montaña real

```

clear
clc
a=imread('US_DEM2016.tif');
B = imresize(a,1/30);
%B=B/10;
figure(1);
surf(B);
demcmap(B,10000);
figure(2);
surf(a)
[a,b]=size(B);

Z2=fliplr(B);
h=130;
W=zeros(a,b,h);
for i=1:a
    for j=1:b
        for k=1:h
            if k>=round(Z2(i,j))
                W(i,j,k)=1;
            end
        end
    end
end
end
end

```

```

W(:,1,:)=0;
W(1,:,:)=0;%izda
W(:,b,:)=0;
W(a,:,:)=0; %dcha

W(:,:h)=0;
W3=double(bwdist(W));

[X, Y, Z] = meshgrid(1:199,1:162,1:130);
[faces, vertices] = MarchingCubes(X, Y, Z, W3, .5);
model_filename = 'real.stl';
stlwrite(model_filename, faces, vertices);
model = createpde();
importGeometry(model, model_filename);
mesh = generateMesh(model, 'GeometricOrder', 'linear');
p = model.Mesh.Nodes;
t = model.Mesh.Elements;
t(end + 1, :) = 1;
displayMesh(p, t);

triangulation_object = triangulation(t(1:end - 1, :)', p');
faces = triangulation_object.freeBoundary';
face_centers_X = (p(1, faces(1, :)) + p(1, faces(2, :)) + p(1,
faces(3, :)))/3;
x_min = min(face_centers_X);
x_max = max(face_centers_X);
inlet_faces = face_centers_X < x_min + 0.1;
outlet_faces = face_centers_X > x_max - 0.1;
e = [faces; ones(3, size(faces, 2))];
e(6, :) = 0;
e(4, inlet_faces) = 2;
e(4, outlet_faces) = 3;
viewCase

[p,e,t,nVnodes,nPnodes,indices] = convertMeshToSecondOrder(p,e,t);

displayMesh(p,t);

% Define fluid
nu = 1.5e-5;

% Init solution and convergence criteria

[u, convergence] = initSolution(p,t,[20 0 0],0);

maxres = 1e-3;
maxiter = 10;
alpha = 0.7;

% Compute wall distance and duplicate velocity vector
d = computeWallDistance(p,e,t,1);
us = u;

% Define inlet velocity profile
vel = [20 0 0];

```

```

% Iterate nonlinear terms
for iter = 1:maxiter
    % Compute turbulent viscosity
    nut = turbulentViscosity(p,t,u,d,nu,indices,'ChenXu');

    % Under-relax velocity and assemble matrix and right hand side
    u = alpha*u+(1-alpha)*us;
    [NS, F] =
assembleNavierStokesMatrix(p,e,t,nu+nut,u(indices.indu),u(indices.indv
),u(indices.indw),'supgDoublyAsymptotic');

    % Apply boundary conditions
    [NS, F] = imposeCfdBoundaryCondition(p,e,t,NS,F,2,'inlet', vel);
    [NS, F] = imposeCfdBoundaryCondition(p,e,t,NS,F,1,'wall');

    % Compute and plot residuals
    [stop, convergence] =
computeResiduals(NS,F,u,size(p),convergence,maxres);
    plotResiduals(convergence,2);

    % Break if solution converged
    if(stop)
        break;
    end

    % Solve equations
    us = u;
    %u = NS\F;
    [u,convergence] =
iterativeSolver(NS,F,u,indices,convergence,'Krylov');
end

vmag = sqrt(u(indices.indu).^2 + u(indices.indv).^2 +
u(indices.indw).^2);

% Plot solutio
figure(3);
displaySolution(p,t,u(indices.indu),'x-velocity');

vx=u(indices.indu)
vy=u(indices.indv)
vz=u(indices.indw)
velocidad=[vx;vy;vz]

points=p'
% Generate the grid of points where the interpolated values will be
calculated
[xg,yg,zg] = meshgrid(0 : 10 : 150);

% Create the interpolating object
F = scatteredInterpolant(points, vx);
% Do the interpolation
uu = F(xg,yg,zg);
% Now you can use isosurface with the gridded data
% Create the interpolating object
G = scatteredInterpolant(points, vy);
% Do the interpolation
v = G(xg,yg,zg);
% Now you can use isosurface with the gridded data
% Create the interpolating object

```

```

H = scatteredInterpolant(points, vz);
% Do the interpolation
w = H(xg,yg,zg);
% Now you can use isosurface with the gridded data
[sx,sy,sz] = meshgrid(0,10:30:140,19:5:23);
figure
surf(Z2)
demcmmap(Z2,10000)
hold on
plot3(sx(:),sy(:),sz(:),'*r');
streamline(xg,yg,zg,uu,v,w,sx,sy,sz)
hold off

```

A.3 Código usado en las pruebas con Forthofer

```

%% load the sample data
%% load the sample data
clear

p1 = -9.3649e-19
p2 = -1.3323e-15
p3 = -8.8229e-14
p4 = 4.7968e-10
p5 = 4.3153e-08
p6 = -9.3411e-05
p7 = 0.0024668
p8 = 11.814

[X] = meshgrid(-800:10:399);
z = p1*X.^7 + p2*X.^6 + p3*X.^5 + p4*X.^4 + p5*X.^3 + p6*X.^2 + p7*X +
p8
[a,b]=size(z);
h=40
W=zeros(a,b,h);
Z2=(z);

figure;
surf(Z2)

for i=1:a
    for j=1:b
        for k=1:h
            if k>=round(Z2(i,j))
                W(i,j,k)=1;
            end
        end
    end
end
W(:,1,:)=0;
W(1,:,:)=0;%izda
W(:,b,:)=0;
W(a,:,:)=0;%dcha

W(:, :, h)=0;
W(:, :, 1)=0;

W3=double(bwdist(W));
[X, Y, Z] = meshgrid(1:120,1:120,1:40);

```

```

[faces, vertices] = MarchingCubes(X, Y, Z, W3, .5);
model_filename = 'peaks_model_2.stl';
stlwrite(model_filename, faces, vertices);
model = createpde();
importGeometry(model, model_filename);
mesh = generateMesh(model, 'GeometricOrder', 'linear');
p = model.Mesh.Nodes;
t = model.Mesh.Elements;
t(end + 1, :) = 1;
displayMesh(p, t);

triangulation_object = triangulation(t(1:end - 1, :)', p');
faces = triangulation_object.freeBoundary';
face_centers_X = (p(1, faces(1, :)) + p(1, faces(2, :)) + p(1,
faces(3, :)))/3;
x_min = min(face_centers_X);
x_max = max(face_centers_X);
inlet_faces = face_centers_X < x_min + 0.1;
outlet_faces = face_centers_X > x_max - 0.1;
e = [faces; ones(3, size(faces, 2))];
e(6, :) = 0;
e(4, inlet_faces) = 2;
e(4, outlet_faces) = 3;
viewCase

[p,e,t,nVnodes,nPnodes,indices] = convertMeshToSecondOrder(p,e,t);

displayMesh(p,t);

% Define fluid
nu = 1.5e-5;

% Init solution and convergence criteria

[u, convergence] = initSolution(p,t,[8.9 0 0],0);
vx=u(indices.indu)'
vy=u(indices.indv)'
vz=u(indices.indw)'
[vx,vy,vz] = meshgrid(3.5,5:10:55,7:5:47);
maxres = 1e-3;
maxiter = 10;
alpha = 0.7;

% Compute wall distance and duplicate velocity vector
d = computeWallDistance(p,e,t,1);
us = u;

% Define inlet velocity profile
vel = [8.9 0 0];

% Iterate nonlinear terms
for iter = 1:maxiter
    % Compute turbulent viscosity
    nut = turbulentViscosity(p,t,u,d,nu,indices,'ChenXu');

    % Under-relax velocity and assemble matrix and right hand side
    u = alpha*u+(1-alpha)*us;

```

```

[NS, F] =
assembleNavierStokesMatrix(p,e,t,nu+nut,u(indices.indu),u(indices.indv
),u(indices.indw),'supgDoublyAsymptotic');

% Apply boundary conditions
[NS, F] = imposeCfdBoundaryCondition(p,e,t,NS,F,2,'inlet', vel);
[NS, F] = imposeCfdBoundaryCondition(p,e,t,NS,F,1,'wall');

% Compute and plot residuals
[stop, convergence] =
computeResiduals(NS,F,u,size(p),convergence,maxres);
plotResiduals(convergence,2);

% Break if solution converged
if(stop)
    break;
end

% Solve equations
us = u;
%u = NS\F;
[u,convergence] =
iterativeSolver(NS,F,u,indices,convergence,'Krylov');
end

vmag = sqrt(u(indices.indu).^2 + u(indices.indv).^2 +
u(indices.indw).^2);

% Plot solutio
figure(3);
displaySolution(p,t,u(indices.indu),'x-velocity');

vx=u(indices.indu)
vy=u(indices.indv)
vz=u(indices.indw)
velocidad=[vx;vy;vz]

points=p'
% Generate the grid of points where the interpolated values will be
calculated
[xg,yg,zg] = meshgrid(3.5 : 1 : 55);
% Create the interpolating object
F = scatteredInterpolant(points, vx);
% Do the interpolation
uu = F(xg,yg,zg);
% Now you can use isosurface with the gridded data

% Create the interpolating object
G = scatteredInterpolant(points, vy);
% Do the interpolation
v = G(xg,yg,zg);
% Now you can use isosurface with the gridded data

% Create the interpolating object
H = scatteredInterpolant(points, vz);
% Do the interpolation
w = H(xg,yg,zg);
% Now you can use isosurface with the gridded data
[sx,sy,sz] = meshgrid(5,10:8:55,17:7:45);
surf(Z2)

```

```

    demcmap(Z2,10000)
hold on
plot3(sx(:),sy(:),sz(:),'*r');
streamline(xg,yg,zg,uu,v,w,sx,sy,sz)
hold off

lims = [0 50 9 25 6 37];
[xsub,ysub,zsub,usub,vsub,wsup] = subvolume(xg,yg,zg,uu,v,w,lims);
figure(5)
    surf(Z2)
    demcmap(Z2,10000)
hold on
streamline(xsub,ysub,zsub,usub,vsub,wsup,sx,sy,sz)
hold off

figure(6)
surf(Z2)
demcmap(Z2,10000)
hold on

% Generate the grid of points where the interpolated values will be
% calculated
[xg,yg,zg] = meshgrid(3.5 : 5: 45);
lims = [10 20 9 25 5 25];
figure(6)
    demcmap(Z2,10000)
    surf(Z2)
    hold on
    % Create the interpolating object
    F = scatteredInterpolant(points, vx);
    % Do the interpolation
    egx = F(xg,yg,zg);
    % Now you can use isosurface with the gridded data

    % Create the interpolating object
    G = scatteredInterpolant(points, vy);
    % Do the interpolation
    egy = G(xg,yg,zg);
    % Now you can use isosurface with the gridded data
    % Create the interpolating object
    H = scatteredInterpolant(points, vz);
    % Do the interpolation
    egz = H(xg,yg,zg);
    [xsub,ysub,zsub,usub,vsub,wsup] =
    subvolume(xg,yg,zg,egx,egy,egz,lims);

quiver3(xsub,ysub,zsub,usub,vsub,wsup,3)

```

A.4 Código usado para comprobar las velocidades

```
px=p(1,:);
py=p(3,:);
[aa,bb]=size(px);
for i=1 : bb
    xaux=px(i);
    yaux=py(i);
    if ((xaux>109) && (xaux<111)) && ((yaux> 12) && (yaux<13))

        ss(i)=1
    end
end

end
```

A.5 Resto de funciones utilizadas

El resto de funciones que se han usado en este TFG no han sido realizadas por el autor de este TFG, por lo que están disponibles en Matlab y en Matlab File Exchange [\[23\]](#).